Function Block
Introduction Guide

Cat. No. R133-IT2-02

Per CX-Programmer Ver. 6.1 (CX-One Ver. 1.1)

Il CD-ROM di CX-One / CX-Programmer include un manuale dell'operatore in formato PDF. Prima di utilizzare il prodotto, è necessario leggere le sezioni "Introduzione", "Precauzioni di sicurezza" e "Modalità d'uso". La guida all'implementazione dei blocchi funzione descrive le operazioni di base per l'utilizzo della Libreria FB di Omron e include suggerimenti per la creazione di un programma utente con i blocchi funzione.
Nella Guida e nel Manuale in formato PDF sono disponibili informazioni dettagliate e avvertenze.
* Per visualizzare il file PDF è necessario disporre di Acrobat Reader 4.0 o versione successiva.

Introduzione

Questo documento fornisce suggerimenti per l'utilizzo della Libreria FB di OMRON e la creazione di blocchi funzione (FB), disponibili per le CPU serie SYSMAC CS1/CJ1-H/CJ1M di Omron (versione 3.0 o successiva) e CX-Programmer Ver. 5.0 o successiva.

Nuove funzioni disponibili in CX-Programmer Ver. 6.0

Nidificazione di FB

È disponibile una funzione di nidificazione degli FB, che semplifica l'utilizzo degli FB come metodo di strutturazione e riutilizzo dei programmi dell'utente. È possibile richiamare gli FB da un programma ST (testo strutturato) ladder, che viene convertito in FB. A tale scopo, sono supportate le seguenti funzioni.

Semplice comprensione della struttura del programma. . . Visualizzatore istanza FB
Gestione dei componenti, inclusi gli FB richiamati. . . Salvataggio e caricamento dei file, inclusi gli FB richiamati
Semplice passaggio agli FB richiamati. . . Possibilità di fare doppio clic sull'istanza FB (istruzione di chiamata)

Monitoraggio del ladder FB

È possibile monitorare il programma ladder FB, analogamente al programma ladder principale.

Finestra a comparsa dei riferimenti incrociati nel ladder FB

Nel programma ladder FB è disponibile una finestra a comparsa dei riferimenti incrociati, analogamente al programma ladder principale.
Utilizzando la barra spaziatrice, è inoltre possibile passare alla bobina di uscita dal contatto.

Passaggio diretto alla guida ST

È possibile andare direttamente all'editor del testo strutturato grazie a un menu a comparsa e visualizzare un argomento della Guida per controllare la sintassi della programmazione ST.

Passaggio diretto al riferimento della libreria FB di Omron

È possibile visualizzare con facilità un file PDF dei riferimenti incrociati della libreria in cui sono descritte le specifiche di una libreria FB di OMRON registrata in un file del progetto.

Attenzione:

benché sia possibile utilizzare un programma che include FB nidificati per una CPU serie CS1/CJ1-H/CJ1M (versione 3.0 o successiva), se si tenta di caricare un programma che contiene FB nidificati di CX-Programmer (Ver. 5.0 o precedente) che non supporta la funzione di nidificazione, non sarà possibile completare l'operazione oppure si otterrà unicamente uno stato incompleto.
Se il file viene salvato così come è, non sarà possibile stabilire la differenza fra il programma incompleto e quello corretto.

[Con CX-Programmer Ver. 5.0]
Al termine del caricamento, verranno visualizzati i seguenti messaggi:

Alcune proprietà PLC non supportate da questa versione di CX-Programmer sono impostate nel PLC di destinazione in collegamento.
Le proprietà del PLC non verranno visualizzate correttamente. Continuare?

[Con CX-Programmer Ver. 4.0]
Al termine del caricamento, verrà visualizzato il seguente messaggio:

Il blocco funzione o altri dati oltre a quelli ladder sono inclusi nei programmi.

[Con CX-Programmer Ver. 3.*]
Al termine del caricamento, viene visualizzato il messaggio "Errore di decompilazione" e non viene visualizzato alcun programma.

Nuove funzioni utilizzabili con CX-Programmer Ver. 6.1

Monitoraggio ST, esecuzione di step

Per facilitare il debug del linguaggio ST di elaborazione sequenziale, sono supportate le seguenti funzioni:

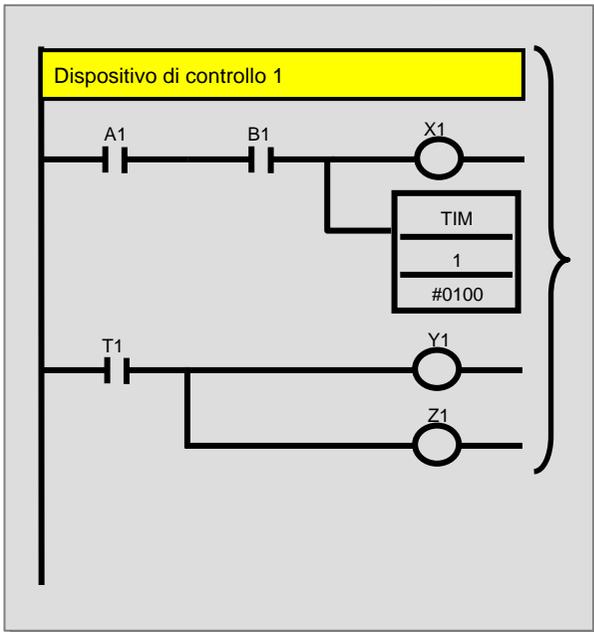
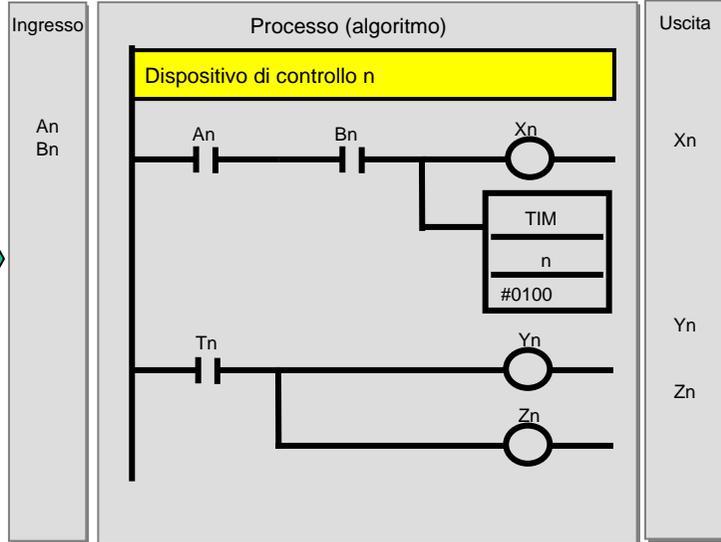
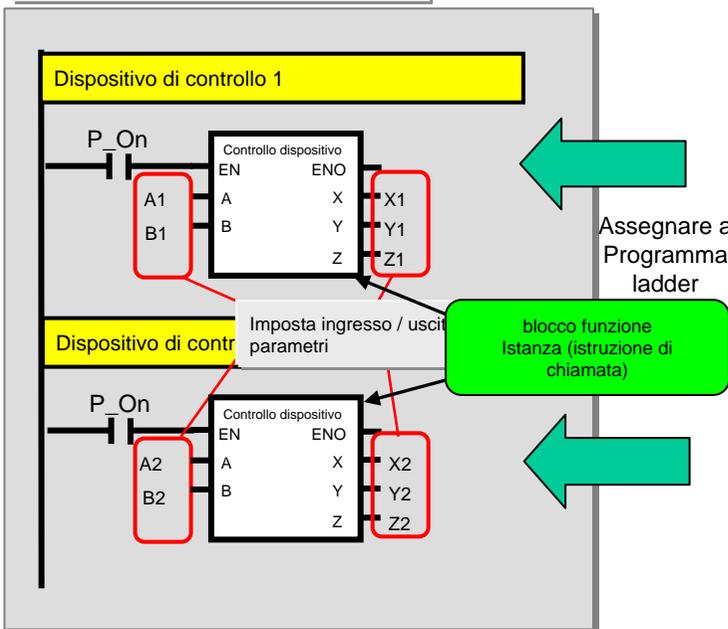
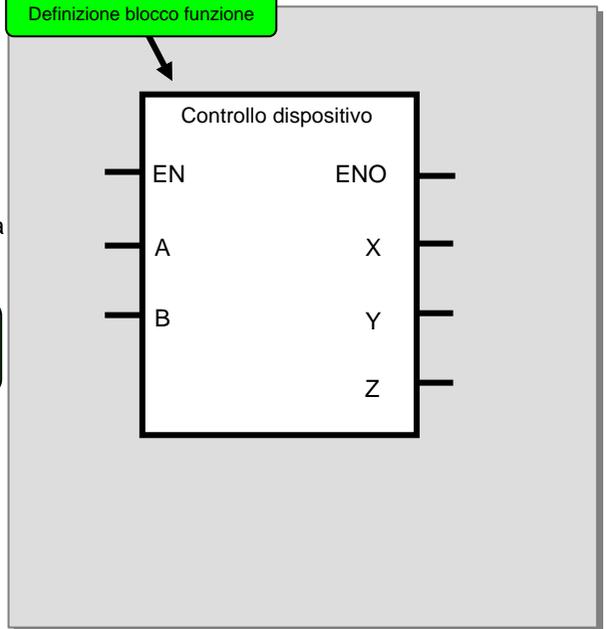
Visualizzazione/modifica del valore corrente durante l'esecuzione del programma ST.
Possibilità di interrompere l'esecuzione in corrispondenza del punto di interruzione e di eseguire gli step con CX-Simulator

Funzioni di protezione FB

È possibile nascondere gli FB che non si desidera vengano visualizzati. In questo modo è possibile proteggere gli FB dalle modifiche erronee, dalle fughe di know-how e dalle modifiche non adeguate al programma.

Capitolo 1
Libreria FB di OMRON

1. Definizione di un blocco funzione

I "Blocchi funzione" sono programmi predefiniti (o funzioni) contenuti all'interno di un singolo elemento di programma e che possono essere utilizzati nel diagramma ladder. Per l'avvio della funzione è necessario un elemento di contatto, ma gli ingressi e le uscite possono essere modificati mediante i parametri utilizzati nella disposizione del ladder.
Le funzioni possono essere riutilizzate come elemento identico (stessa memoria) oppure presentarsi come un nuovo elemento a cui viene assegnata una memoria propria.

Definizione di blocco funzione … Contiene la logica definita (algoritmo) e l'interfaccia I/O. Nella definizione del blocco funzione gli indirizzi in memoria non vengono assegnati.
Istanza del blocco funzione (istruzione di chiamata) ... Si tratta dell'istruzione che richiama l'istanza del blocco funzione quando viene utilizzata dal programma ladder, tramite la memoria assegnata all'istanza

2. Esempio di blocco funzione

Le figure di seguito mostrano un esempio di blocco funzione per un circuito di limitazione del tempo, da utilizzare nel ladder È possibile modificare il set point dell'istruzione TIM per riassegnare il tempo impostato per lo spegnimento dell'uscita nel rung ladder. Utilizzando il blocco funzione nel modo indicato di seguito, è possibile rendere arbitrario il limite di tempo del circuito semplicemente cambiando un parametro specifico.

Diagramma ladder

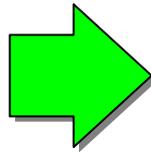
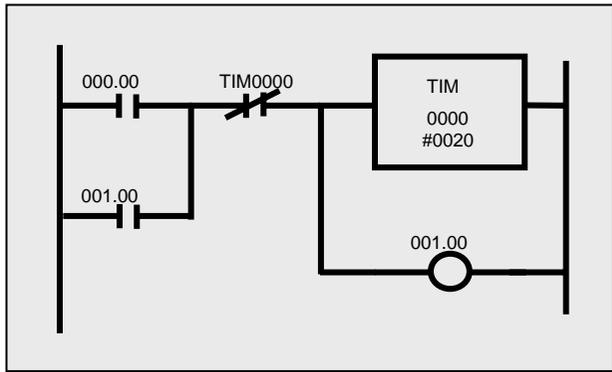
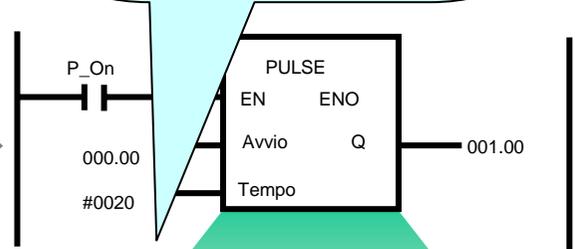
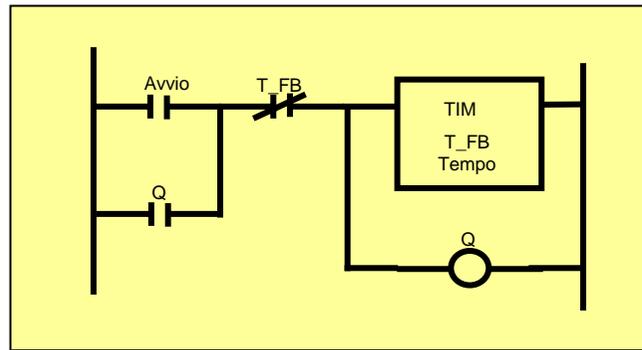
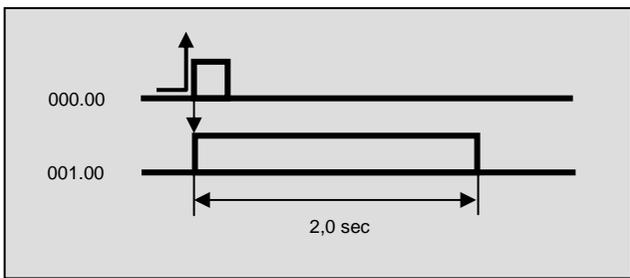



Diagramma di temporizzazione

3. Panoramica della Libreria FB di OMRON.

La libreria FB di OMRON raccoglie i file Blocco funzione predefiniti forniti da Omron. Questi file devono essere utilizzati come supporto per semplificare la programmazione e contengono funzionalità standard per la programmazione delle funzioni relative ai componenti PLC e FA di Omron.

3-1. Vantaggi della Libreria FB di OMRON.

La libreria FB di OMRON è una raccolta di esempi di blocchi funzione che ha lo scopo di migliorare la connettività dei moduli dei PLC e dei componenti FA prodotti da Omron. Di seguito viene fornito un elenco dei vantaggi offerti dalla libreria FB di OMRON:

(1) Nessuna necessità di creare diagrammi ladder con le funzioni di base dei moduli PLC e dei componenti FA

È possibile dedicare più tempo ai programmi su misura per i dispositivi esterni, anziché creare diagrammi ladder di base, poiché questi risultano già disponibili.

(2) Semplicità di utilizzo

Per ottenere un programma funzionale è necessario caricare il file blocco funzione in modo che venga eseguita la funzionalità desiderata, quindi immettere un'istanza (istruzione di chiamata dei blocchi funzione) nel programma del diagramma ladder e infine impostare gli indirizzi (parametri) per gli ingressi e le uscite.

(3) Non è necessario verificare il funzionamento del programma

Omron ha eseguito i test sulla libreria blocco funzione. Non è necessario che l'utente esegua il debug dei programmi per utilizzare il modulo e i componenti FA dei PLC.

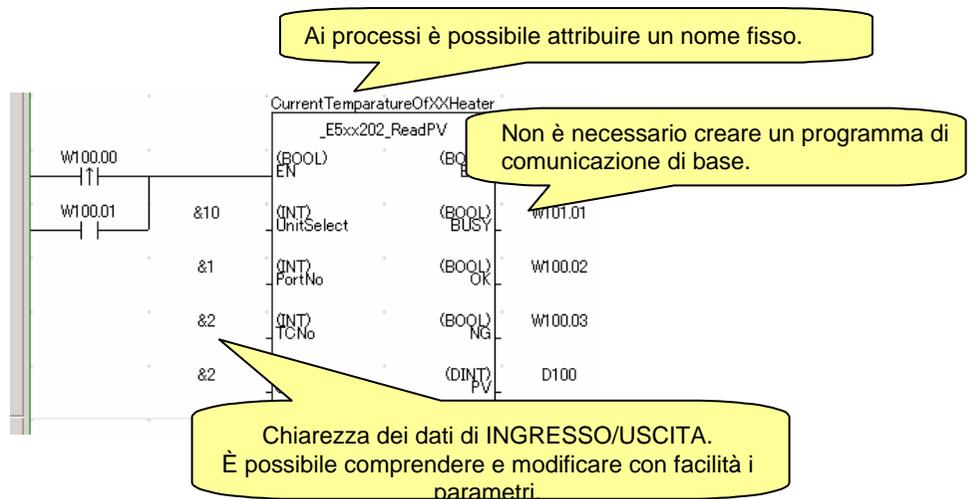
(4) Semplicità di comprensione

Nel blocco funzione, il nome del corpo e delle istanze è chiaramente visualizzato. Al processo è possibile attribuire un nome fisso.

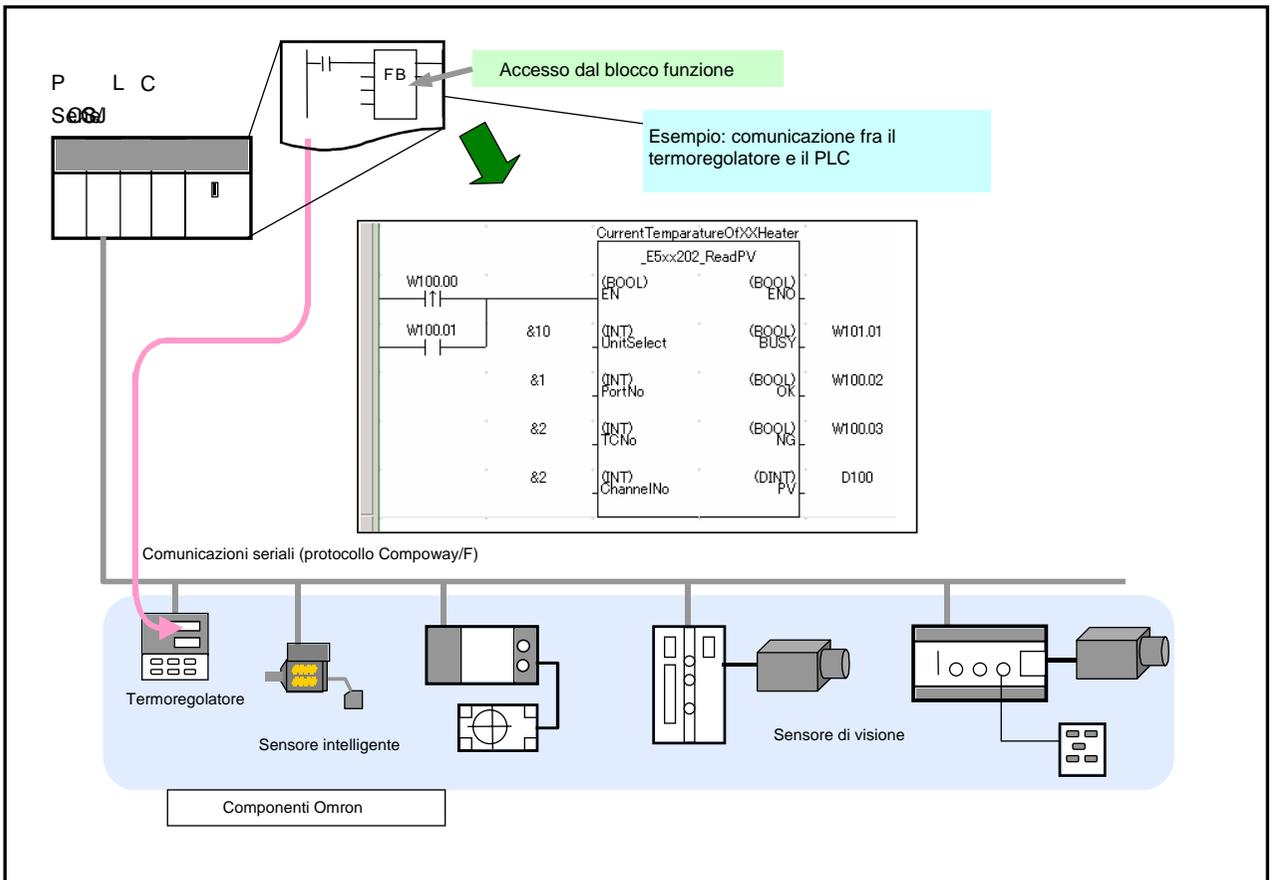
L'istanza (l'istruzione di chiamata del blocco funzione) ha parametri di ingresso e uscita. Poiché i dati del relè temporaneo e di elaborazione non sono visualizzati, i valori degli ingressi e delle uscite risultano più visibili. Inoltre, dato che viene localizzata la modifica dei parametri, il controllo durante il debug e altre operazioni risulta più semplice.

Infine, poiché l'elaborazione interna del blocco funzione non viene visualizzata quando l'istanza viene utilizzata nel diagramma ladder, il programma del diagramma ladder risulta più comprensibile all'utente finale.

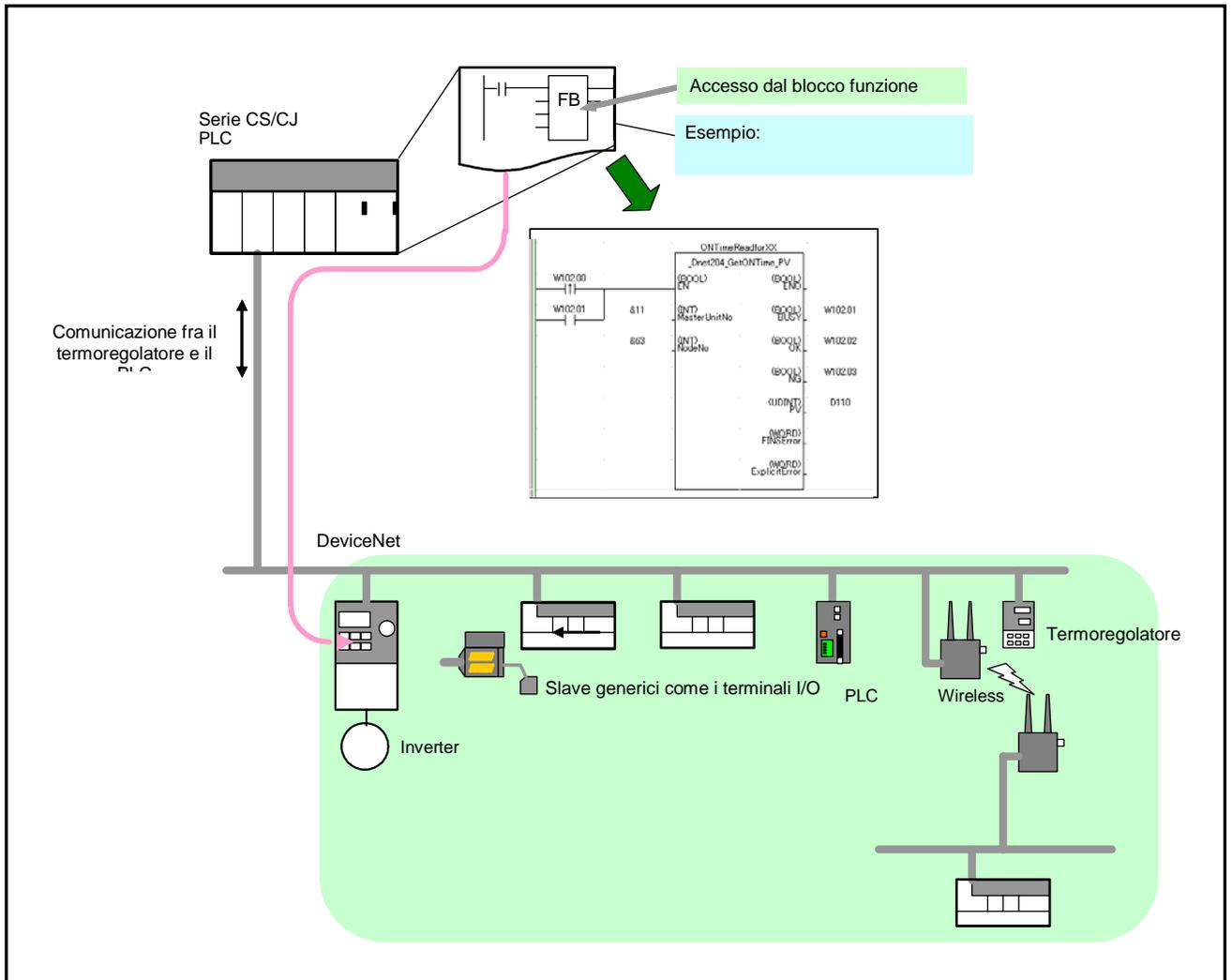
(5) Espandibilità futura

Omron non modificherà l'interfaccia fra il diagramma ladder e i blocchi funzione. In caso di aggiornamenti dei PLC e dei componenti FA, sarà possibile utilizzare i moduli sostituendo il blocco funzione con quello corrispondente per il nuovo modulo e ottimizzare le prestazioni.

3-2-1. Esempio di utilizzo della Libreria FB di OMRON - 1

È possibile controllare con facilità i componenti predefiniti di Omron dal diagramma ladder del PLC.

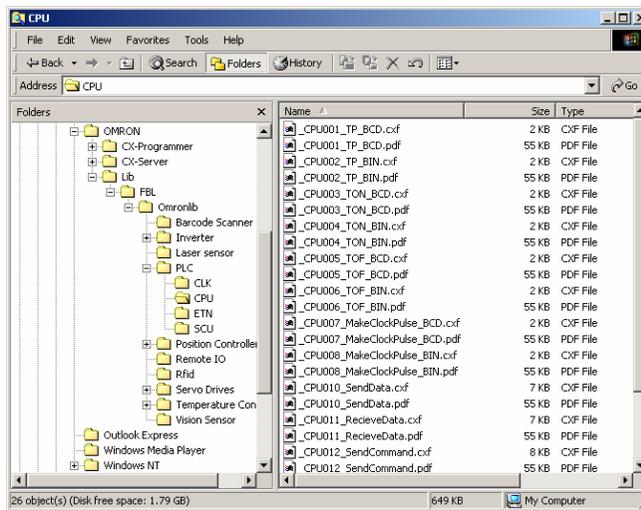
- Possibilità di configurare comunicazioni a basso costo (RS-232C/485)

3-2-2. Esempio di utilizzo della Libreria FB di OMRON - 2

Con il livello DeviceNet è possibile ottenere comunicazioni ad alte prestazioni.

- Possibilità di stabilire con facilità comunicazioni fra il PLC e gli slave DeviceNet.

3-3. Contenuto della Libreria FB di OMRON

Di seguito sono indicati i componenti della libreria FB di OMRON:

3-3-1. File della parte OMRON FB

Il file della parte OMRON FB viene preparato con il blocco funzione del diagramma ladder per definire ciascuna funzione del modulo PLC e del componente FA.
I file contengono un programma scritto nel diagramma ladder e hanno l'estensione .CXF.
Il nome del file della parte OMRON FB inizia con "_" (carattere di sottolineatura).
Quando la libreria FB di OMRON viene installata in un personal computer, i file della parte OMRON FB vengono inseriti nella cartella appropriata di ciascun modulo PLC e componente FA nella directory di installazione Omron.

3-3-2. Riferimento alla libreria

Il riferimento alla libreria descrive le specifiche di funzionamento del file della parte OMRON FB e di quelle dei relativi parametri di ingresso e uscita. Questo file è in formato PDF.
Quando si utilizza la libreria FB di OMRON, è necessario selezionare il file della parte OMRON FB, impostare i parametri di ingresso/uscita e verificare il funzionamento del programma utilizzando il riferimento alla libreria.

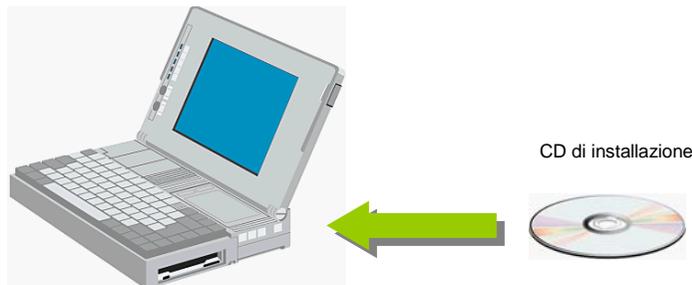
V60x 200	Read Data Carrier Data _V60x200_ReadData
FB name	_V600_ReadData
Symbol	_V60x200_ReadData Start trigger ↑, Busy Flag → (BOOL) EN Unit No. → (INT) Unit No Head No. → (INT) HeadNo Data Carrier read address → (WORD) CarrierAddress Bytes to read from Data Carrier → (WORD) ReadBytes Read data storage area type → (WORD) StoreAreaID Read data storage word address → (INT) StoreAreaNo Communications designation → (INT) Communications Processing designation → (INT) ByteOrder Cancel → (BOOL) Cancel (BOOL) ENO → Busy Flag (BOOL) BUSY → Normal end (BOOL) OK → Error end (BOOL) NG → Error code (May be omitted.) (WORD) ErrorCode
File name	¥Lib¥FBL¥English¥omronlib¥RFID¥V600¥_V60x200_ReadData10.cxf
Applicable models	CS1W-V600C11/V600C12 and CJ1W-V600C11/V600C12 ID Sensor Units
Basic function	Reads data from a Data Carrier.
Conditions for usage	Other • This FB cannot be executed if the ID Sensor Unit is busy. The NG Flag will turn ON if an attempt is made.
Function description	Data is read from the specified area of the Data Carrier specified by the *Unit No.* and *Vendor No.* Up to 2048 bytes (1024 words) can be read at one time. The word designation for storing the data is specified using the area type and beginning word address. For example, for D1000, the area type is set to P_DM and the beginning word address is set to &1000.
EN input condition	Connect EN to an OR between an upwardly differentiated condition for the start trigger and the BUSY output from the FB.
Restrictions Input variables	• Always use an upwardly differentiated condition for EN. • If the input variables are out of range, the ENO Flag will turn OFF and the FB will not be processed. • Always specify a head number of &1 for One-Head ID Sensor Units (CS1W-V600C11 and CJ1W-V600C11).

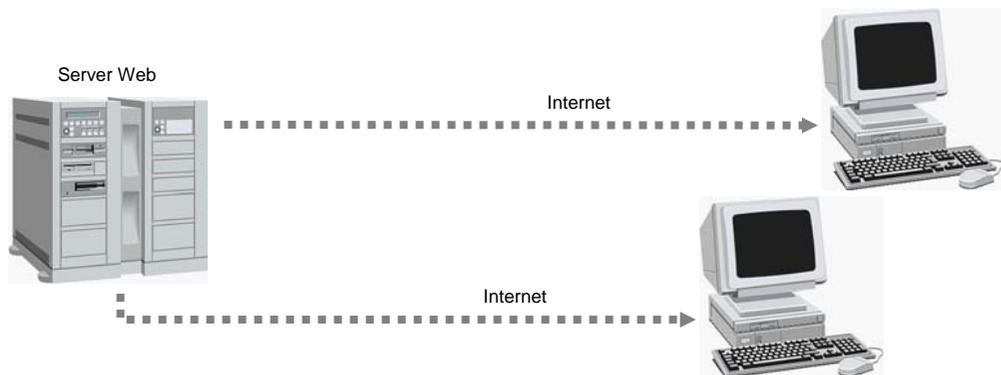
3-4. Catalogo file e punti di accesso alla Libreria FB di OMRON

3-4-1. Catalogo dei file della libreria FB di OMRON

Tipo	Componenti di destinazione	Numero di file della parte OMRON FB (aggiornato al febbraio 2005)
Componenti FA	Termoregolatore, sensore intelligente, sensore di identificazione, sensore di visione, lettori di codici a barre bidimensionali, terminale wireless	circa 80
PLC	CPU, scheda di memoria, modulo di I/O CPU speciale (Ethernet, Controller Link, modulo DeviceNet, termoregolatore)	circa 95
Componenti controllo assi	Modulo di posizionamento Invertitore Servoazionamento	circa 70

3-4-2. CD di installazione di CX-One / CX-Programmer

La libreria FB di OMRON è disponibile nel CD di installazione di CX-One / CX-Programmer.
È possibile installarla durante l'installazione di CX-One / CX-Programmer.

3-4-3. Accesso ai file della libreria FB di OMRON dal server Web

Omron ha messo a disposizione nel server Web la versione più recente dei file della libreria FB di OMRON. Verranno aggiunti nuovi file per il supporto di moduli PLC e componenti FA nuovi o migliorati.
Per scaricare la libreria FB di OMRON è sufficiente selezionare il relativo menu dal sito Web di Omron. Questa funzione è disponibile in tutti i paesi.

Capitolo 2
Utilizzo della Libreria FB di OMRON

1. Definizione del programma di destinazione

In questo capitolo viene descritto come utilizzare la libreria FB di Omron con il file della parte OMRON FB "Make ON Time/OFF Time Clock Pulse in BCD".

1-1. Specifiche dell'applicazione

Di seguito sono riportate le specifiche dell'applicazione di destinazione:-
- L'impulso viene generato dopo che la modalità PLC è passata da "run" a "monitor" o viceversa.
- L'uscita dell'impulso avviene sull'indirizzo 1.00.
- Il tempo di attivazione dell'impulso generato viene impostato su D100.
- Il tempo di disattivazione dell'impulso generato è di 2 secondi.

1-2. Specifiche del file della parte OMRON FB

Di seguito sono riportate le specifiche del file della parte OMRON FB "Make ON Time/OFF Time Clock Pulse in BCD":-

CPU 007	Make ON Time/OFF Time Clock Pulse in BCD _CPU007_MakeClockPulse_BCD
Basic function	Generates a clock pulse with the specified ON time and OFF time and outputs it to ENO.
Symbol	Always ON (P_On); ONtime (unit: 100 ms); OFFtime (unit: 100 ms); _CPU007_MakeClockPulse_BCD (BOOL) EN (BOOL) ENO (WORD) OnTime (WORD) OffTime
File name	¥Lib¥FBL¥English¥omronlib¥PLC¥CPU¥_CPU007_MakeClockPulse_BCD10.cxf
Applicable models	CS1-H, CS1-H, and CJ1M CPU Units
Conditions for usage	PLC Properties • The PV update method for timers and counters must be set to BCD in the PLC Setup. A compiling error will occur if BCD mode is not set. The mode can be set in the PLC Properties in the CX-Programmer. PLC Properties: General | Protection | Function Block; Name: 新規PC1; Type: CS1G-H CPU44; Verify; Mode: Program, Debug, Monitor, Run; ☑ Use comment instructions; ☑ Use section markers; ☑ Display dialog to show PLC Memory Backup Status; ☑ Use IR/DRs independently per task; ☐ Execute Timer/Counter as Binary Shared Resources • Timers
Function description	ENO will be OFF for the time set in *OFF time* and then will be ON for the time set in *ON time*. EN ON/OFF; ENO ON/OFF; OnTime (*100 ms); OffTime (*100 ms)
EN input condition	Connect the EN input to the Always ON Flag (P_On).
Restrictions Input variables	• If the input variables are out of range, the ENO Flag will turn OFF and the FB will not be processed. • Set the *ON time* and *OFF time* input variables to between #0000 and #9999 in BCD (100 ms units). If a setting is not within range, ENO is turned OFF.
Application example	In the following example, bit A will be repeatedly ON for 5 s and OFF for 3 s. Always ON (P_On); ONtime (unit: 100 ms) #50; OFFtime (unit: 100 ms) #30; _CPU007_MakeClockPulse_BCD (BOOL) EN (BOOL) ENO (WORD) OnTime (WORD) OffTime; Bit A
Related FBs	Use the correct FB for the timer/counter PV update mode set in the PLC Setup. Binary mode: Make ON Time/OFF Time Clock Pulse in Binary (_CPU008_MakeClockPulse_BIN) BCD mode: Make ON Time/OFF Time Clock Pulse in BCD (_CPU007_MakeClockPulse_BCD)

■ **Variable Tables**

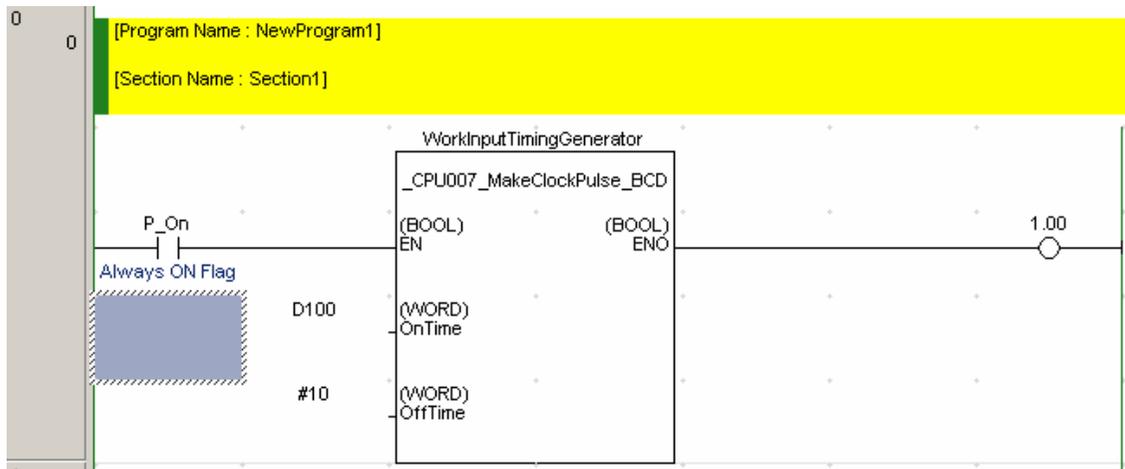
Input Variables

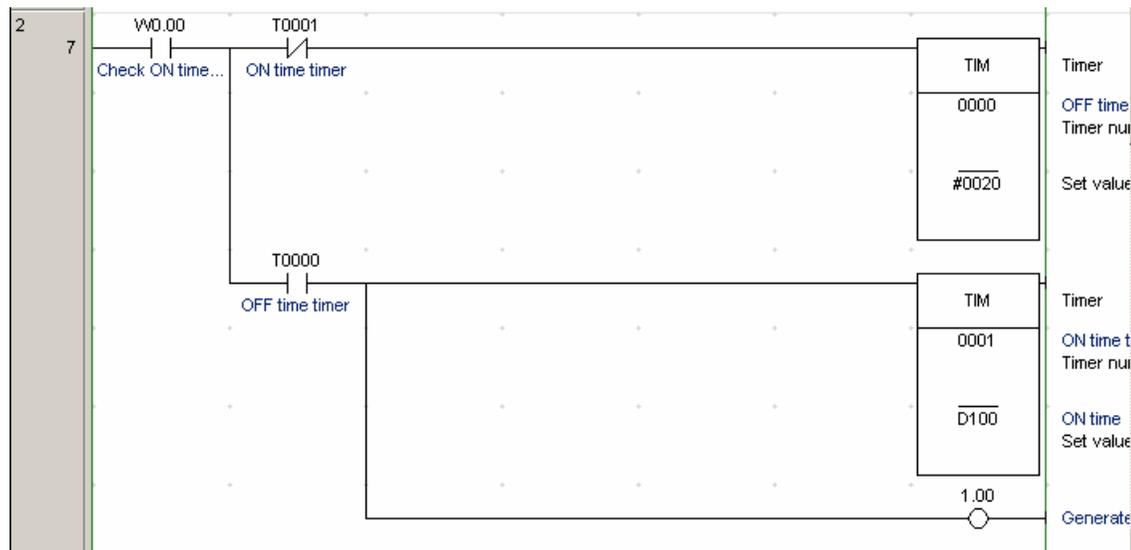
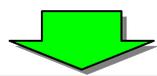
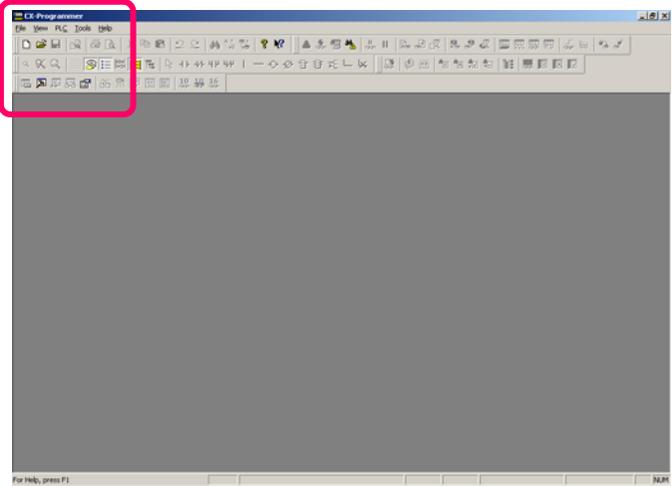
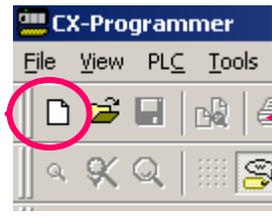
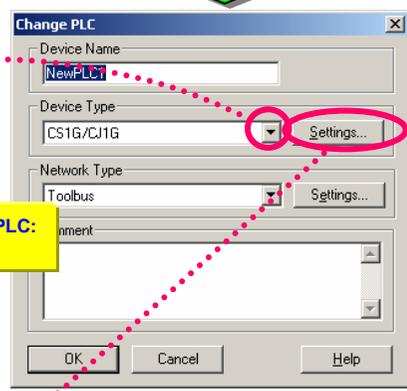
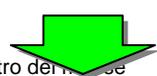
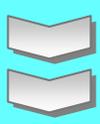
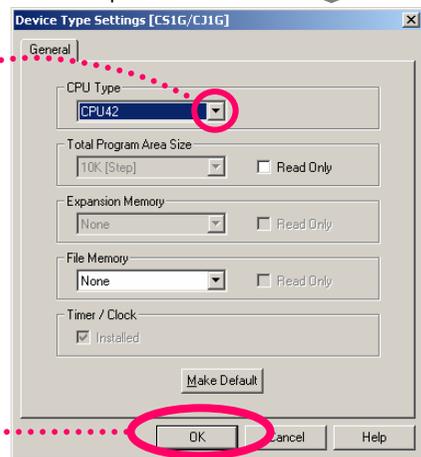
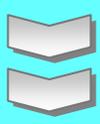
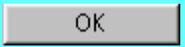
Name	Variable name	Data type	Default	Range	Description
EN	EN	BOOL			1 (ON): FB started 0 (OFF): FB not started.
ON time	OnTime	WORD		#0000 to #9999	Specify the ON time (unit: 100 ms). For example, #30 means 3 seconds.
OFF time	OffTime	WORD		#0000 to #9999	Specify the OFF time (unit: 100 ms). For example, #30 means 3 seconds.

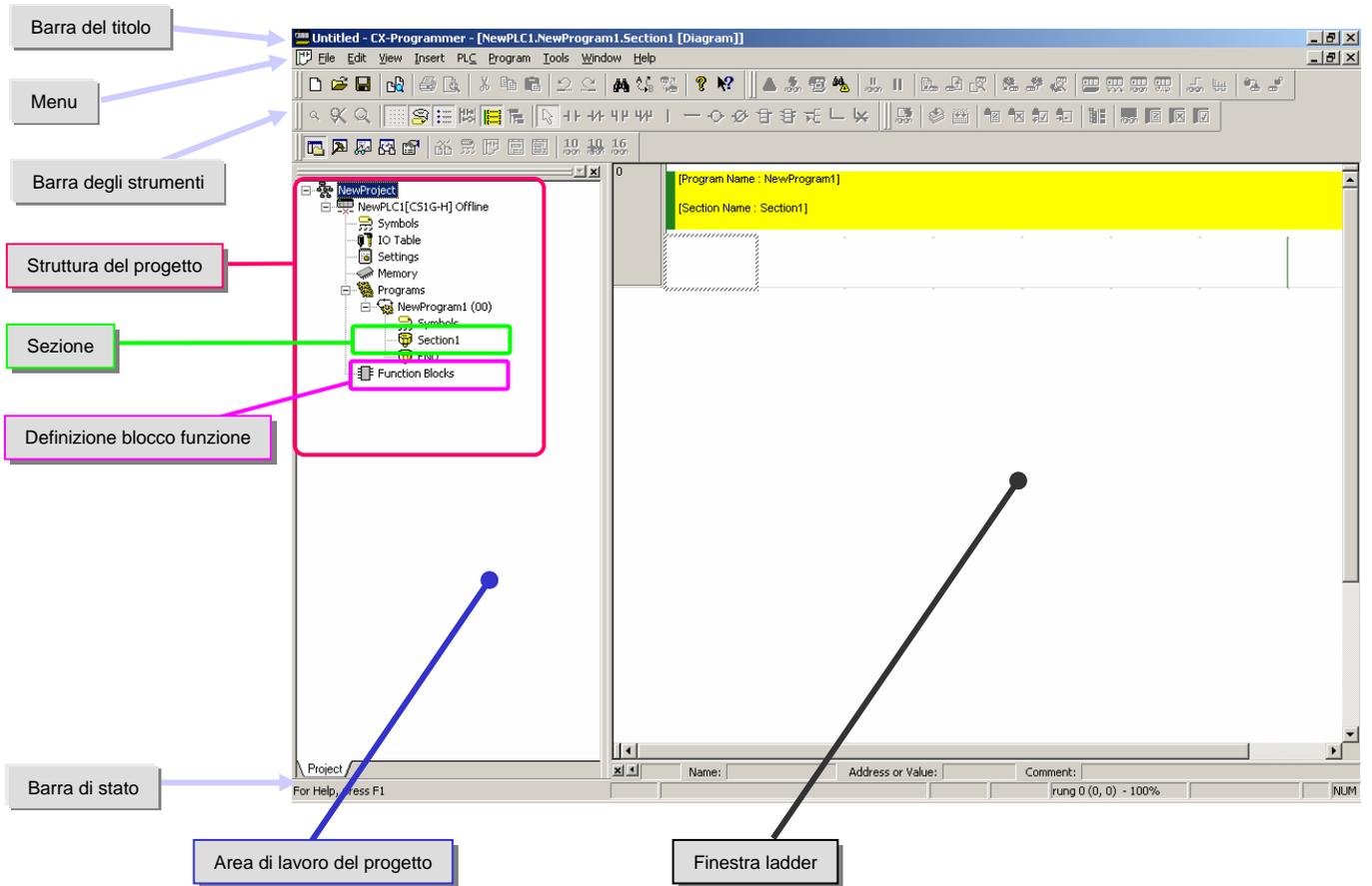
Output Variables

Name	Variable name	Data type	Range	Description
ENO	ENO	BOOL		Turns ON for the OnTime and OFF for the OffTime.

1-3. Immissione del programma

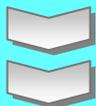
Creare il seguente programma ladder:-

[Riferimento] Se viene creato come un diagramma ladder diretto, il programma avrebbe il seguente aspetto:-

inizione del programma di des
rtura di un nuovo progel
Importazione della libreria FB
eazione di un programm
ontrollo dei programmi
2. Apertura di un nuovo progetto e impostazione del tipo di dispositivo
Fare clic sul pulsante [Nuovo] nella barra degli strumenti di CX-Programmer.
Fare clic su
CX-Programmer
File View PLC Tools
Change PLC
Device Name
NewPLC1
Device Type
CS1G/CJ1G
Settings...
Network Type
Toolbus
Comment
OK
Cancel
Help
Fare clic con il pulsante sinistro del mouse.
! Per utilizzare i blocchi funzione, selezionare i seguenti PLC:
CS1G-H, CS1H-H, CJ1G-H, CJ1H-H, CJ1M
Settings...
Fare clic con il pulsante sinistro del mouse
Device Type Settings [CS1G/CJ1G]
General
CPU Type
CPU42
Total Program Area Size
10K [Step]
Read Only
Expansion Memory
None
File Memory
Timer / Clock
Installed
Make Default
Fare clic per selezionare un tipo di CPU.
OK
Fare clic su [OK] per stabilire il tipo di CPU selezionato.

3. Funzioni della finestra principale

In questa sezione vengono illustrate le varie funzioni della finestra principale.

Nome	Contenuto/Funzione
Barra del titolo	Mostra il nome del file dei dati salvati creato in CX-Programmer.
Menu	Consentono di selezionare le voci di menu.
Barre degli strumenti	Consente di selezionare le funzioni facendo clic sulle icone. Selezionare [Visualizza] -> [Barre degli strumenti] per visualizzare le barre degli strumenti. Per modificare le posizioni di visualizzazione, è possibile trascinare le barre degli strumenti.
Sezione	Consente di suddividere un programma in più blocchi, che possono essere creati e visualizzati singolarmente.
Area di lavoro del progetto Struttura del progetto	Controlla i programmi e i dati. Consente di copiare i dati degli elementi trascinandoli e incollandoli fra progetti diversi oppure all'interno dello stesso progetto.
Finestra ladder	Schermata per la creazione e la modifica di un programma ladder.
Definizione blocco funzione	Mostra la definizione del blocco funzione. Selezionando le icone, è possibile copiare o eliminare la definizione blocco funzione selezionata. - L viene visualizzato nel caso di un file della parte OMRON FB. - Nel caso di un blocco funzione definito dall'utente, L e visualizzato per der oppure S per ST.
Barra di stato	Mostra informazioni quali il nome del PLC, lo stato in linea/non in linea o la posizione della cella attiva

4. Importazione del file della parte OMRON FB

Con il cursore, selezionare l'icona della definizione del blocco funzione dalla struttura del progetto e fare doppio clic su di essa. Selezionare Inserisci blocco funzione, quindi spostarsi su un file della libreria e selezionarlo.

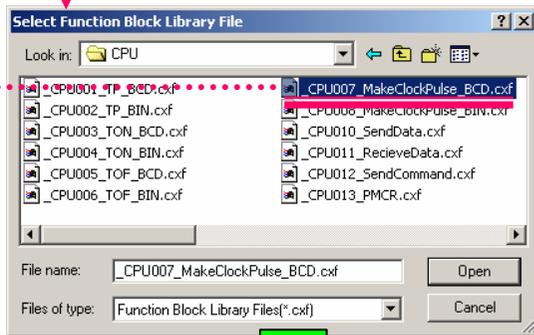
Fare clic con il pulsante destro del mouse
® Inserisci blocco funzione
® File della libreria

Fare doppio clic con il pulsante sinistro del mouse.
® [OmronLib]
® [Controllore programmabile]
® [CPU]
Effettuare le selezioni precedenti nell'ordine indicato.

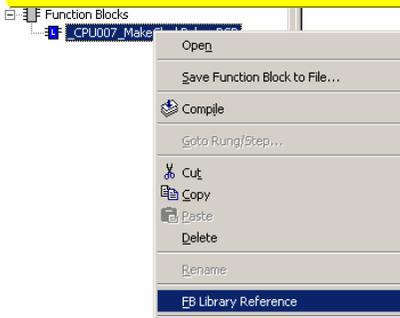
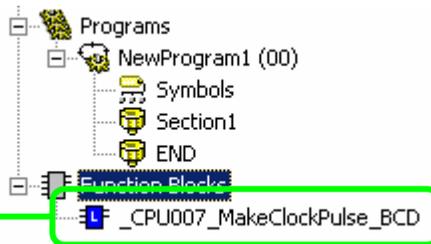
Fare doppio clic con il pulsante sinistro del mouse

Nella finestra di dialogo "Selezionare il file della libreria blocco funzione", selezionare il file della parte OMRON FB desiderato.

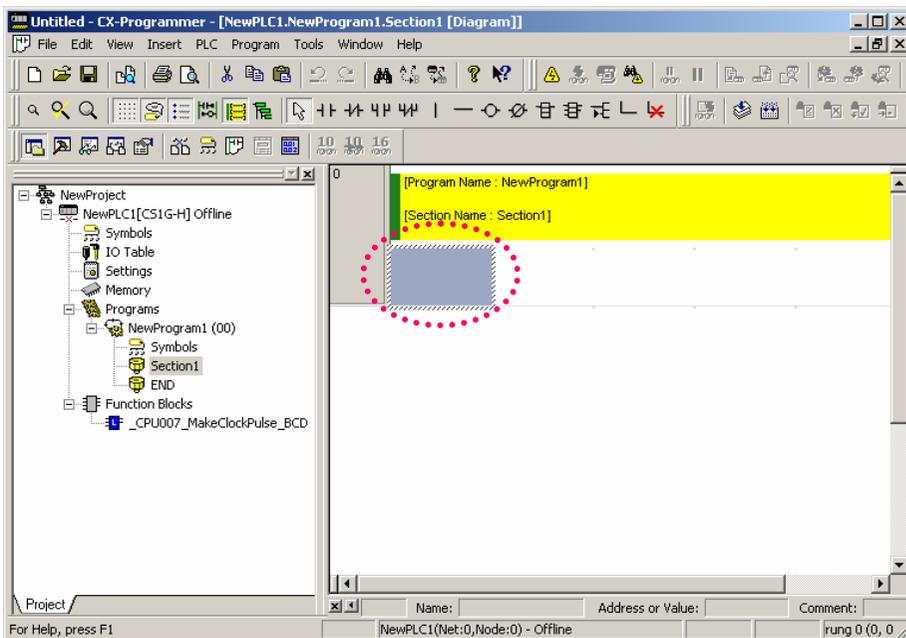
! Il percorso predefinito della libreria FB di OMRON è
C:\Programmi\Omron\CX-One\Lib\FBL.

Fare clic su "_CPU007_MakeClockPulse_BCD.cxf"

Fare clic sul pulsante [Apri]

! Per controllare con facilità le specifiche dei file della parte OMRON FB, selezionare i file della parte OMRON FB registrati e [Riferimento libreria FB] da un menu a comparsa, quindi visualizzare un file di riferimento della libreria.

La definizione del blocco funzione "_CPU007_MakeClockPulse_BCD" viene registrata come parte del file del progetto.

Definizione blocco funzione

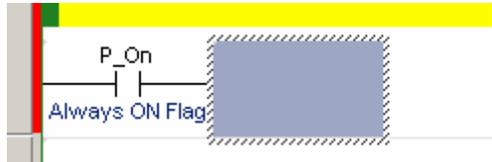
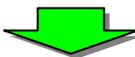
5. Creazione di programmi

Verificare la posizione del cursore in alto a sinistra della finestra ladder per iniziare la programmazione.

5-1. Immissione di un contatto normalmente aperto

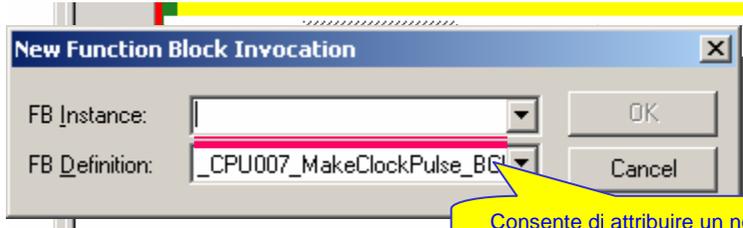
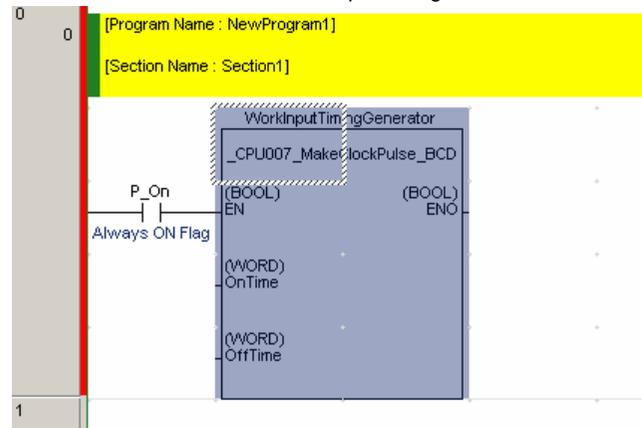
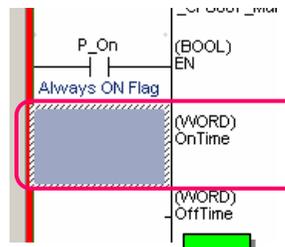
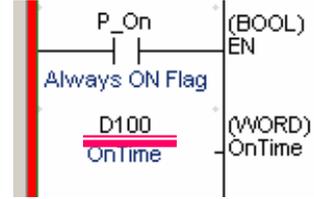
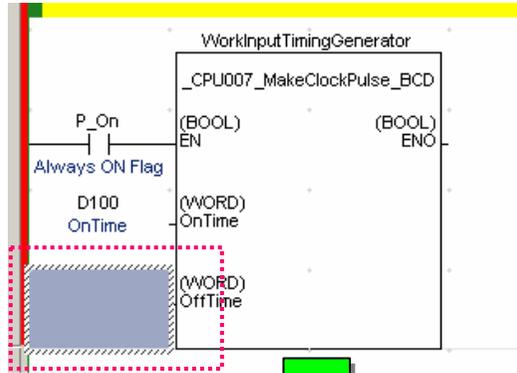
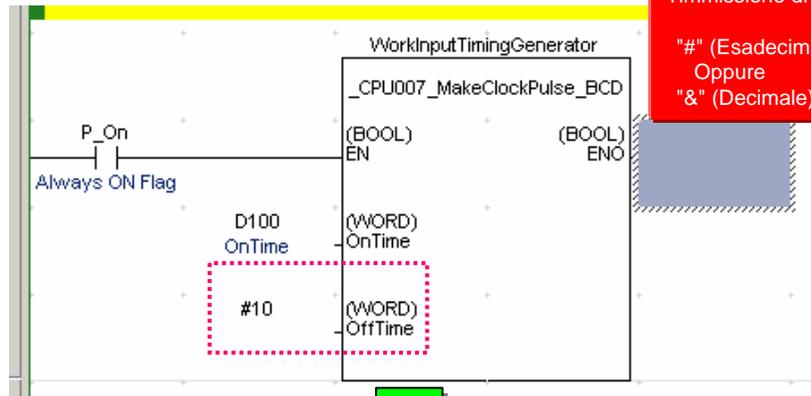
C

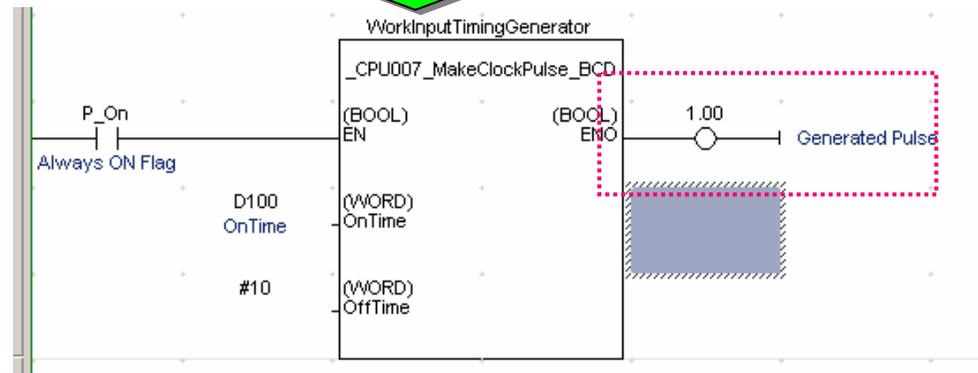
Premere [C] sulla tastiera per aprire la finestra di dialogo [Nuovo contatto].
Selezionare il simbolo "P_On" dalla casella di riepilogo.

P_On

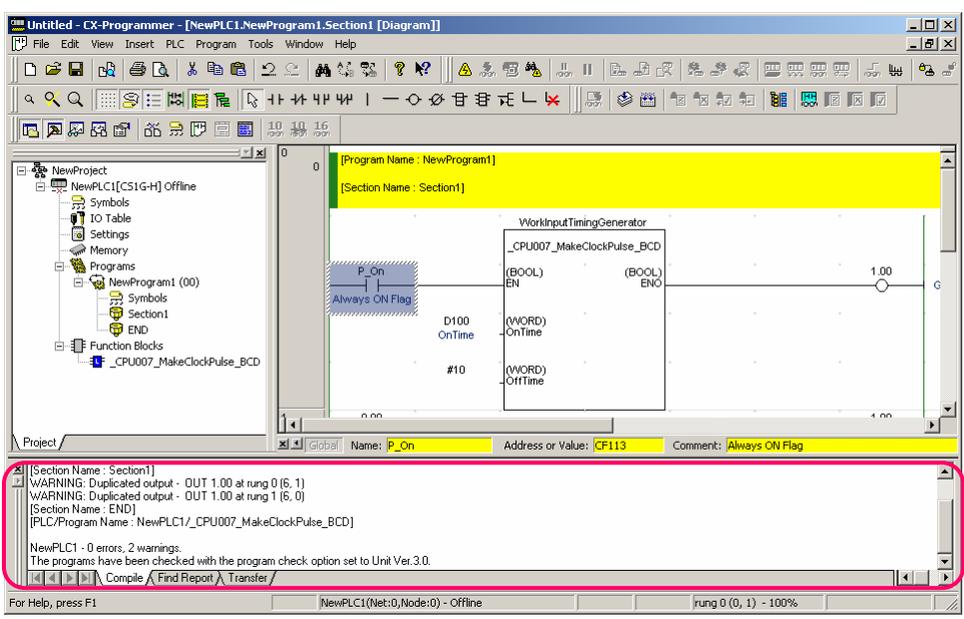
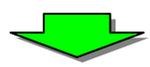
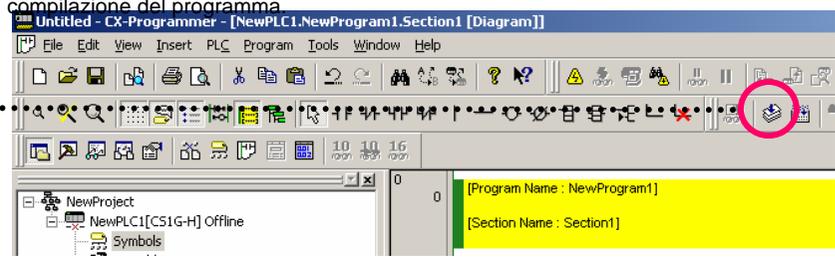
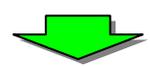
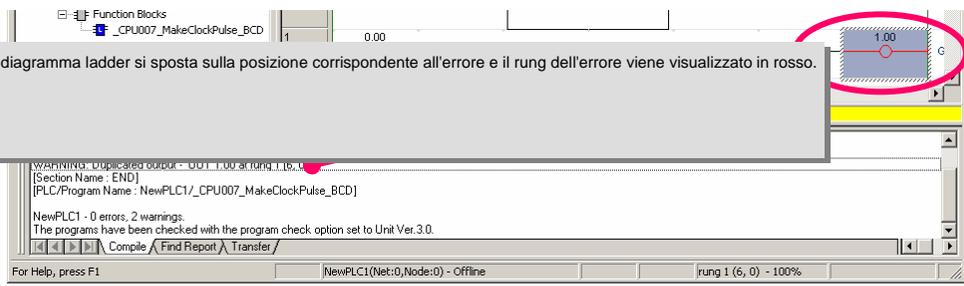
ENT

Eliminazione di comandi
- Spostare il cursore sul comando e premere il tasto CANC oppure
- Spostare il cursore sulla cella destra del comando e premere il tasto Backspace.

"P_On" è un simbolo definito dal sistema. Il relativo stato è sempre ON.
Lo 0 della cifra superiore di un indirizzo viene omesso.
Fra il numero del canale e il numero di relè viene inserito un carattere [.] (punto).

inizione del programma di desi
rtura di un nuovo proge
Importazione della libreria FB
eazione di un programm
ontrollo dei programmi
5-2. Immissione di un'istanza
F
Premere il tasto [F] della tastiera per aprire la finestra di dialogo [Nuova chiamata di blocco funzione].
New Function Block Invocation
FB Instance:
FB Definition:
_CPU007_MakeClockPulse_BC
OK
Cancel
Consente di attribuire un nome al processo specifico nel diagramma.
Immettere il testo per creare il nome dell'istanza FB.
[WorkInputTimingGenerator]
ENT
Mostra l'istruzione di chiamata FB "WorkInputTimingGenerator".
[Program Name : NewProgram1]
[Section Name : Section1]
WorkInputTimingGenerator
_CPU007_MakeClockPulse_BCD
P_On
Always ON Flag
(BOOL)
EN
(BOOL)
ENO
(WORD)
OnTime
(WORD)
OffTime
5-3. Immissione di parametri o ENT
P
Spostare il cursore a sinistra del parametro di ingresso
New Parameter
d100
Detail >>
OK
Cancel
Immettere l'indirizzo.
[d100]
ENT
D100
OnTime
Scegliere l'indirizzo del parametro di ingresso "OnTime".

Immettere i parametri restanti utilizzando la stessa procedura.

Aggiungere il seguente prefisso per l'immissione di costanti come parametri:

"#" (Esadecimale/BCD)
Oppure
"&" (Decimale)

6. Verifica degli errori di programma (Compila)

Prima di trasferire il programma, verificare la presenza di eventuali errori utilizzando la funzione di compilazione del programma.

Fare clic

Gli errori e gli indirizzi sono visualizzati nella finestra dei risultati.

Fare doppio clic sugli errori visualizzati. Il cursore del diagramma ladder si sposta sulla posizione corrispondente all'errore e il rung dell'errore viene visualizzato in rosso.

Modificare l'errore.

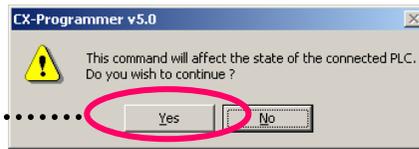
- Durante la verifica del programma, viene automaticamente visualizzata la finestra dei risultati.
- Premendo il tasto J o F4, il cursore si sposta sulla posizione dell'errore.
- Premere il tasto [Esc] per chiudere la finestra dei risultati.

7. Collegamento in linea

In linea per trasferimen → Monitoraggio → Modifica in linea

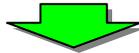
8. Monitoraggio - 1

Impostare la modalità Monitor del PLC (Simulatore).

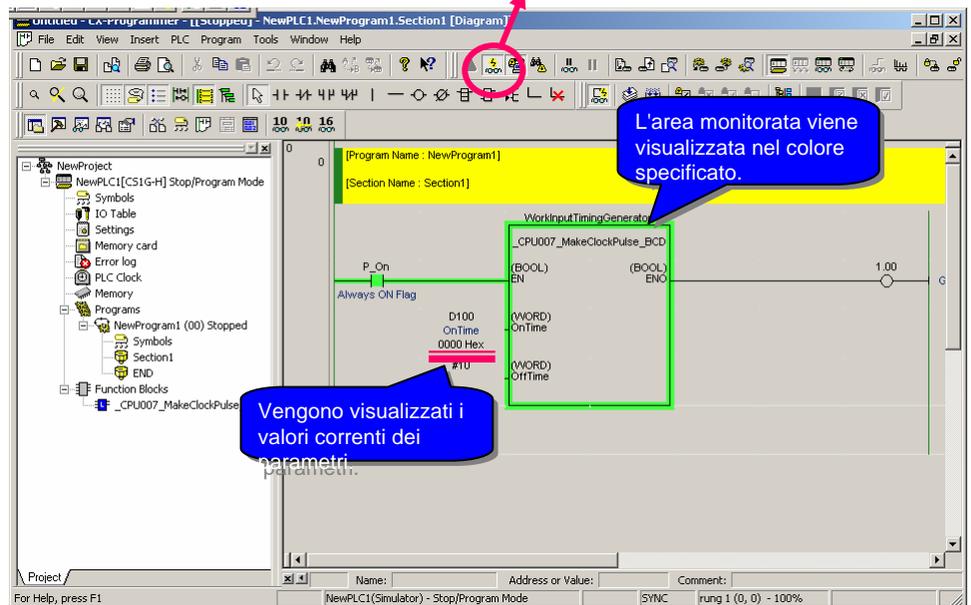
È possibile monitorare lo stato di attivazione/disattivazione dei contatti e delle bobine.

Fare clic su

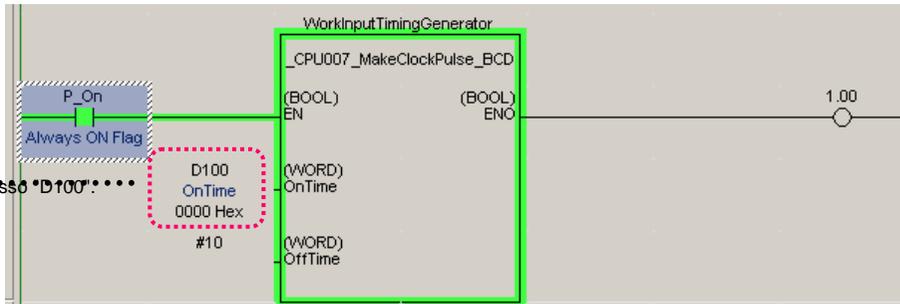
Fare clic su [Sì].

Se il programma include un volume ingente di dati, la velocità di scorrimento dello schermo potrebbe risultare rallentata durante il monitoraggio.

Per ovviare a tale problema, fare clic sull'icona seguente per annullare il monitoraggio, spostarsi sull'indirizzo da monitorare e attivare nuovamente la modalità Monitor.

— attiva/disattiva il monitoraggio del PLC.

L'area monitorata viene visualizzata nel colore specificato.

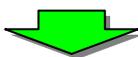
Vengono visualizzati i valori correnti dei parametri.

9. Monitoraggio - 2 Modifica del valore corrente dei parametri

Modificare il valore corrente dei dati dei contatti/bobine o canali nella finestra ladder.

Spostare il cursore sul parametro di ingresso "D100".

Fare clic con il pulsante destro del mouse e selezionare la voce di menu.

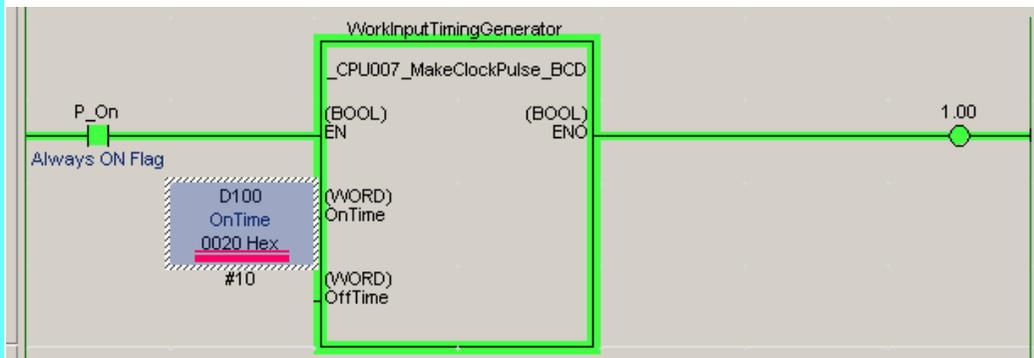
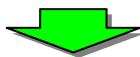
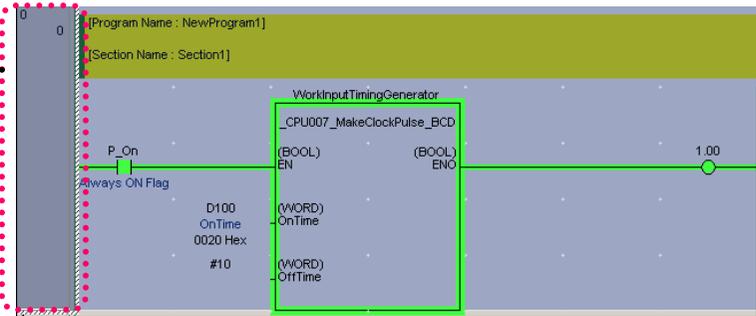
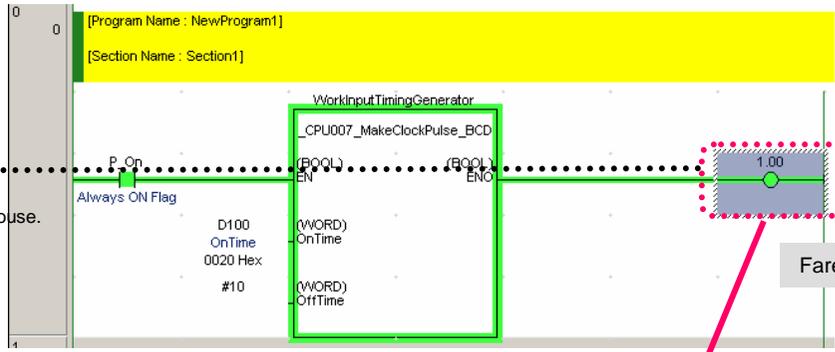
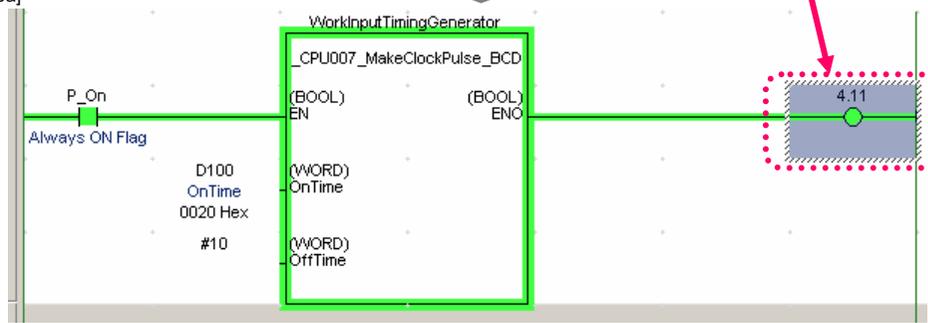
[Imposta/Reset(S)]
® [Valore impostazione (V)]

Oppure

Fare doppio clic con il pulsante sinistro del mouse.

Oppure

Modificare il valore corrente del parametro di ingresso.

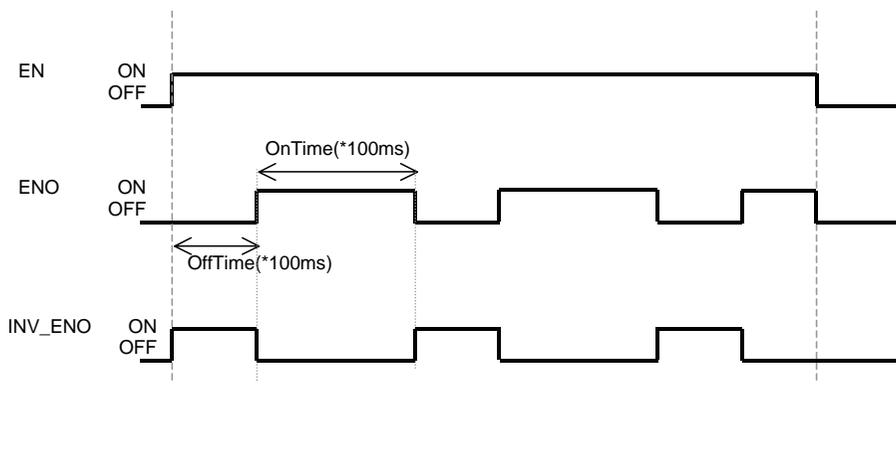
Fare clic su [Imposta]

Aggiungere il seguente prefisso per l'immissione di costanti come parametri:
"#" (Esadecimale/BCD)
Oppure
"&" (Decimale)

Operazione in linea
In linea per trasferimenti
Monitoraggio
Modifica in linea
10. Modifica in linea
Spostare il cursore sul rung da modificare.
Per selezionare più rung, utilizzare la funzione di trascinamento del mouse.
[Program Name : NewProgram1]
[Section Name : Section1]
WorkInputTimingGenerator
_CPU007_MakeClockPulse_BCD
P_On
Always ON Flag
(BOOL) EN
(BOOL) ENO
1.00
D100
OnTime
0020 Hex
#10
(WORD) OnTime
(WORD) OffTime
Selezionare [Programma] ® [Modifica in linea] ® [Inizia]
Tasto di scelta rapida: [Ctrl]+[E]
Spostare il cursore sulla bobina da modificare.
Fare doppio clic con il pulsante sinistro del mouse.
Fare doppio clic
Modificare l'indirizzo con il numero di bit necessario (4.11 nell'esempio)
-( )- Edit Coil
4.11
Detail >>
OK
Cancel
Selezionare [Programma] ® [Modifica in linea] ® [Invia modifiche]
Tasto di scelta rapida: [Ctrl]+[Maiusc]+[E]
4.11
Fine

Capitolo 3
Personalizzazione del file del componente OMRON FB

1. Definizione del programma di destinazione

In questo capitolo viene descritto come personalizzare la libreria FB di Omron con il file della parte OMRON FB "Make ON Time/OFF Time Clock Pulse in BCD".

1-1. Modifica delle specifiche del file

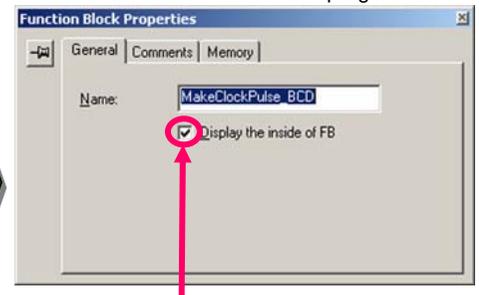
Il file della parte OMRON FB "Make ON Time/OFF Time Clock Pulse in BCD" è stato sviluppato per disattivare ripetutamente l'ENO per il tempo di disattivazione (OffTime) specificato (unità: 100 msec) e attivarlo per il tempo di attivazione (OnTime) specificato (unità: 100 msec). In questo esempio, il file della parte OMRON FB verrà modificato in modo da produrre un segnale di inversione aggiungendo il parametro di uscita "INV_ENO".

1-2. Modifica del contenuto del file della parte OMRON FB

Per soddisfare i requisiti indicati sopra, è necessario apportare le seguenti modifiche al file della parte OMRON FB "Make ON Time/OFF Time Clock Pulse in BCD"

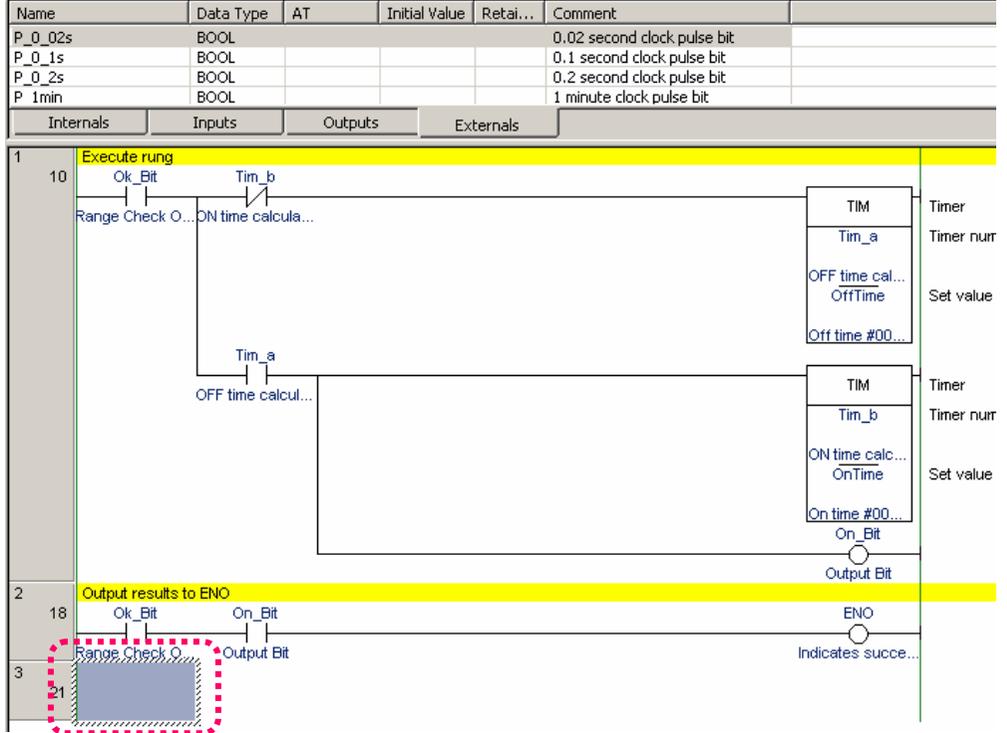
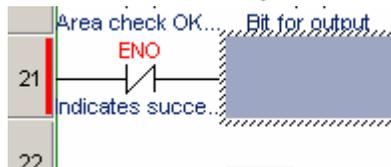
1. Aggiungere il parametro di uscita "INV_ENO".
2. Aggiungere il programma ladder per produrre l'ENO e invertire il segnale.

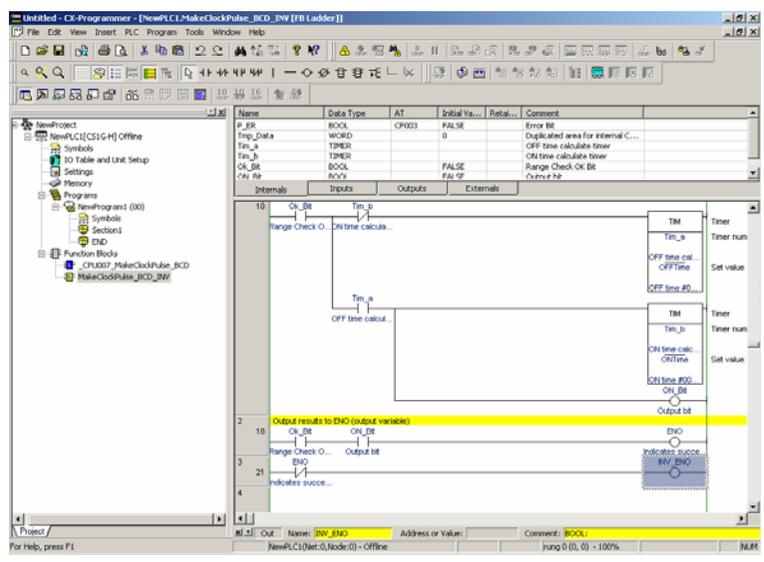
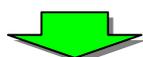
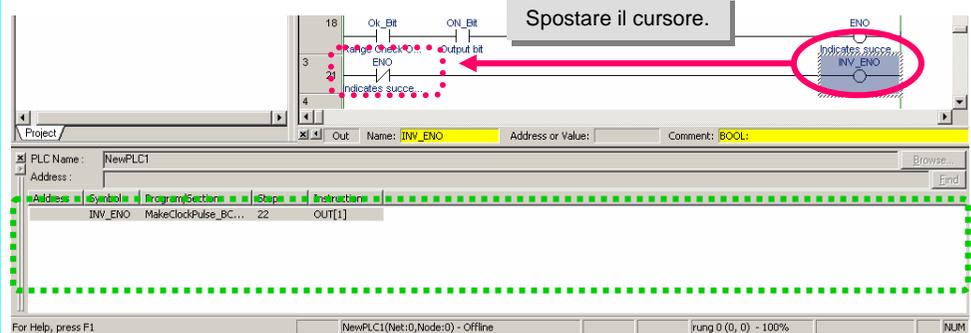
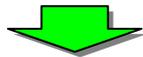
Attenzione

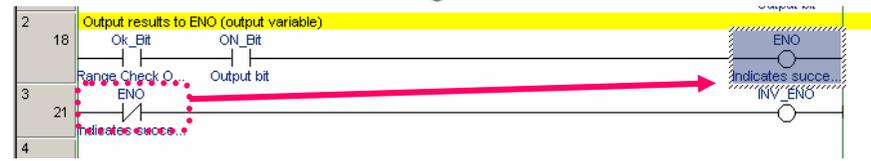
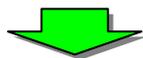
OMRON non garantisce il funzionamento delle parti OMRON FB personalizzate. Verificare accuratamente il funzionamento della parte FB prima di procedere con la personalizzazione e successivamente confermare il funzionamento di ciascuna parte FB.

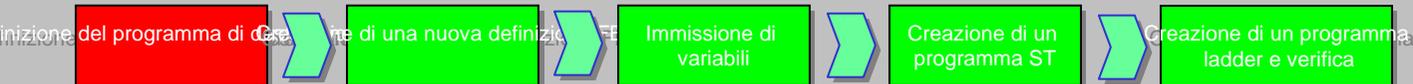
inizione del programma di design
Copia della parte FB
Modifica della definizione FB
2. Copiare il file della parte OMRON FB
Importare il file della parte FB "Make ON Time/OFF Time Clock Pulse in BCD" come illustrato nel capitolo 1 (nome definizione FB: _CPU007_MakeClockPulse_BCD)
NewProject
NewPLC1[CS1G-H] Offline
Symbols
IO Table
Settings
Memory
Programs
NewProgram1 (00)
Section1
END
Function Blocks
_CPU007_MakeClockPulse_BCD
Selezionare l'icona della parte OMRON FB quindi fare clic con il pulsante destro del mouse.
® Copia
Selezionare l'icona della definizione blocco funzione e fare clic con il pulsante destro del mouse.
® Incolla
Copy_CPU007_MakeClockPulse_BCD
Il file della parte OMRON FB viene incollato.
Modificare il nome della definizione FB.
Selezionare l'icona del blocco funzione incollata e fare clic con il pulsante destro del mouse.
® Rinomina
[MakeClockPulse_BCD_INV]
MakeClockPulse_BCD_INV
Nota:
L'utente non può creare le definizioni blocco funzione il cui nome inizia con "_" (carattere di sottolineatura). Utilizzare nomi che non iniziano con "_".
Attivare la modifica del codice del programma FB inter
Function Block Properties
General
Comments
Memory
Name:
MakeClockPulse_BCD
Display the inside of FB
Selezionare l'icona del blocco funzione incollata e fare clic con il pulsante destro del mouse.
® Proprietà
Oppure
ALT
+
ENT
Fare clic sulla casella di controllo.

inizione del programma di desi
Copia della parte FB
difica della definizione FB
3. Aggiunta di una variabile al blocco funzione
Tabella variabili
Aprire l'Editor ladder del blocco funzione.
Untitled - CX-Programmer - [NewPLC1.MakeClockPulse_BCD_INV [FB Ladder]]
Consente di aprire l'Editor ladder del blocco funzione.
Selezionare l'icona del blocco funzione con il cursore del mouse e fare doppio clic con il pulsante sinistro del mouse.
Name
Data Type
AT
Initial Value
Retai...
Comment
P_ER
BOOL
CF003
Error flag
Tmp_Data
WORD
0
Multiple area of use for internal...
Tim_a
TIMER
OFF time measurement timer
Tim_b
TIMER
ON time measurement timer
Ok_Bit
BOOL
FALSE
Area check OK flag
On_Bit
BOOL
FALSE
Bit for output
Internals
Inputs
Outputs
Externals
Setting value check
Circuit execution
BCD To Binary
Source word
Result word
Timer
Timer number
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
MakeClockPulse_BCD
MakeClockPulse_BCD_INV
Nel file originale della parte OMRON FB è anche possibile visualizzare il relativo programma ladder, ma non modificarlo.
Editor ladder
Tabella variabili
Name
Data Type
AT
Initial Value
Retai...
Comment
ENO
BOOL
FALSE
Indicates successful execution ...
Con il cursore del mouse, selezionare la scheda Uscite nella Tabella variabili
Quindi fare clic con il pulsante sinistro del mouse.
Internals
Inputs
Outputs
Externals
Fare clic e selezionare Inserisci variabile(I).
Immettere il nuovo nome della variabile.
New Variable
Name:
INV_ENO
Data Type:
BOOL
Usage:
Output
Initial Value:
FALSE
Retain
Comment:
Inverting output of ENO
OK
Cancel
Advanced...
Selezionare BOOL come dati bit.
Verificare la correttezza della variabile immessa.
Name
Data Type
AT
Initial Value
Retai...
Comment
ENO
BOOL
FALSE
Indicates successful execution ...
INV_ENO
BOOL
FALSE
Inverting output of ENO

4. Modifica del ladder del blocco funzione

Aggiungere il diagramma ladder richiesto al campo di modifica del ladder del blocco funzione.
Spostare il cursore sulla colonna di sinistra del rung successivo.

4-1. Immissione di un contatto

4-2. Controllo dello stato dell'utilizzo delle variabili

Analogamente al programma ladder principale, è possibile utilizzare una finestra a comparsa dei riferimenti incrociati per controllare l'utilizzo delle variabili.

Visualizzare la finestra a comparsa dei riferimenti incrociati.

Con il cursore del mouse, selezionare LDNOT dalla finestra a comparsa dei riferimenti incrociati.

La variabile ENO viene utilizzata anche in una bobina di uscita nel passo 20.

Il cursore nell'editor ladder FB si sposta sulla bobina di uscita nel passo 20.

Capitolo 4
Utilizzo del linguaggio ST (Testo strutturato)

1. Informazioni sul linguaggio ST

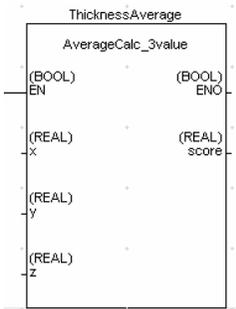
Il linguaggio ST (Testo strutturato) è un codice di alto livello per i sistemi di controllo industriale (principalmente i PLC) definito dallo standard IEC 61131-3.
Include molte istruzioni di controllo, fra cui IF-THEN-ELSE-END_IF, il ciclo FOR / WHILE e molte funzioni matematiche, quali SIN / LOG.
Rappresenta la scelta ideale per l'elaborazione matematica.
Il linguaggio ST supportato da CX-Programmer è conforme allo standard IEC 61131-3.
Di seguito sono riportate le funzioni aritmetiche disponibili in CX-Programmer Ver. 5/6:

seno (SIN), coseno (COS), tangente (TAN), arcoseno (ASIN), arcocoseno (ACOS), arcotangente (ATAN), radice quadrata (SQRT), valore assoluto (ABS), logaritmo (LOG), logaritmo naturale (LN), esponenziale naturale (EXP), esponenziazione (EXPT)

```
(* Initial Settings *)
XMT[1] := 2;
XMT[2] := 7;
N := 2;

(* CRC16 *)
CRCTMP := 16#FFFF;
FOR I := 1 TO N DO
    CRCTMP := CRCTMP XOR XMT[1];
    FOR J := 1 TO 8 DO
        CT := CRCTMP AND 1;
        IF CRCTMP < 0 THEN
            CH := 1;
            CRCTMP := CRCTMP AND 16#7FFF; (* CRCTMP & 0x7FFF *)
        ELSE
            CH := 0;
        END_IF;
        UINT_CRCTMP := WORD_TO_UINT(CRCTMP) / 2;
        CRCTMP := UINT_TO_WORD(UINT_CRCTMP);
        IF CH = 1 THEN
            CRCTMP := CRCTMP OR 16#4000;        (* CRCTMP OR 0x4000 *)
        END_IF;
        IF CT = 1 THEN
            CRCTMP := CRCTMP XOR 16#A001; (* CRCTMP XOR 0xA001 *)
        END_IF;
    END_FOR;
END_FOR;

IF CRCTMP < 0 THEN
    CL := 1;
    CRCTMP := CRCTMP AND 16#7FFF;       (* CRCTMP & 0x7FFF *)

ELSE
    CL := 0;
END_IF;

C_1 := CRCTMP AND 16#FF;                (* CRCTMP & 0xFF *)
CRCTMP := CRCTMP AND 16#7F00;       (* CRCTMP & 0x7F00 *)
UINT_CRCTMP := WORD_TO_UINT(CRCTMP) / 256;
C_2 := UINT_TO_WORD(UINT_CRCTMP);
```

Riferimento: IEC 61131-3 è uno standard internazionale per la programmazione dei controllori programmabili (PLC) definito dalla commissione IEC (International Electro-technical Commission).
Lo standard si compone di 7 parti. La parte 3 definisce la programmazione dei PLC.

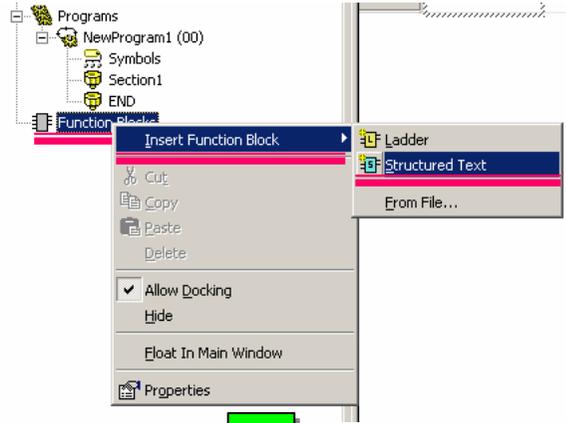
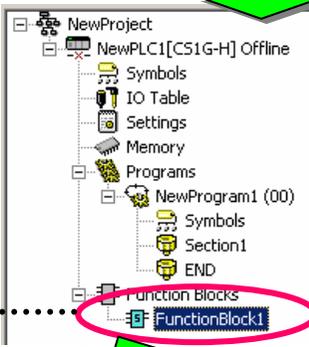
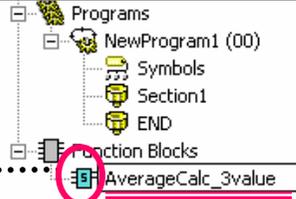
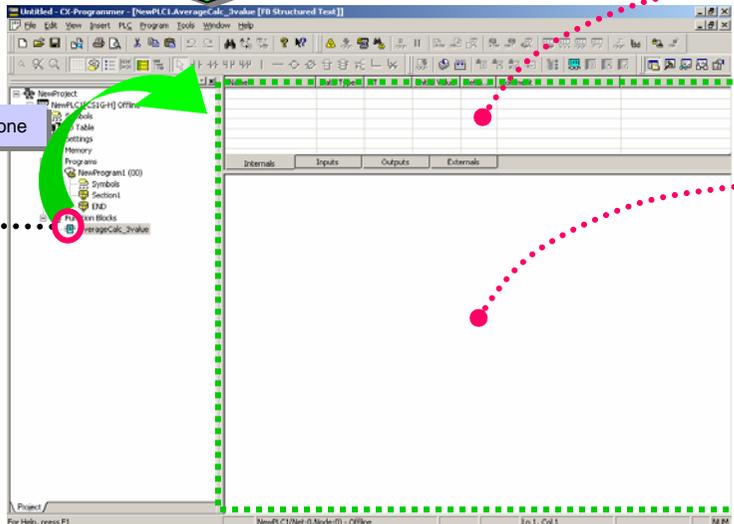
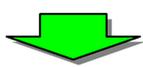
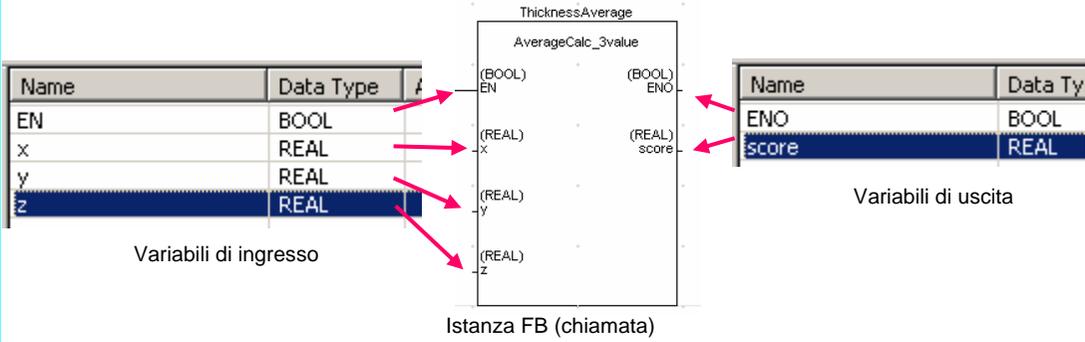
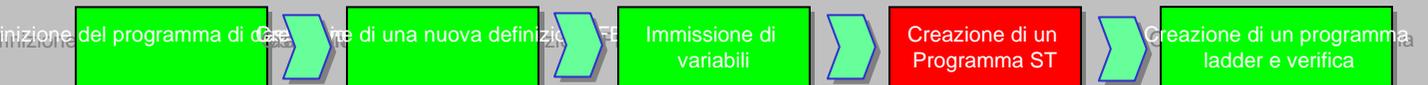
2. Definizione del programma di destinazione

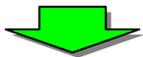
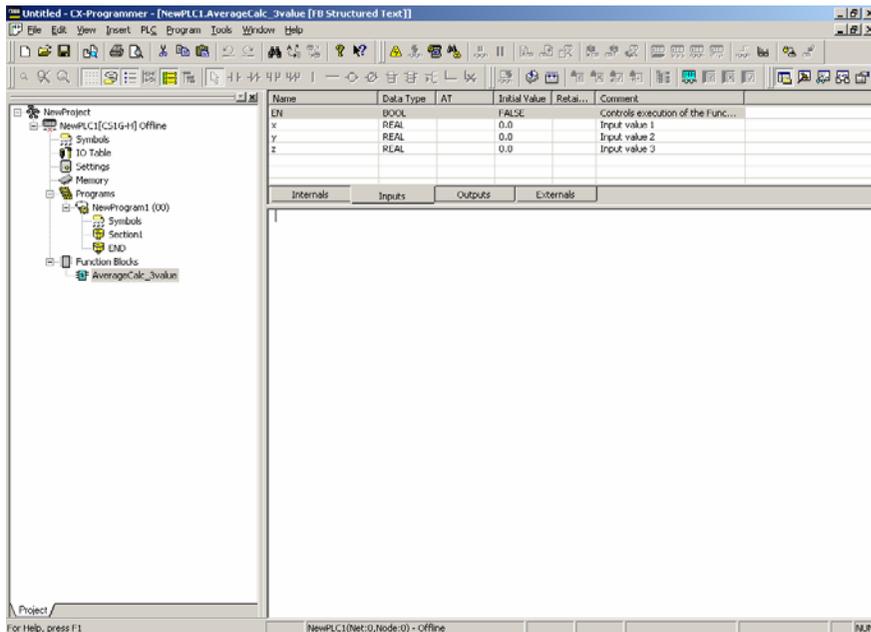
In questo esempio viene descritta la creazione di un programma ST in un blocco funzione per calcolare il valore medio dello spessore misurato.

Per memorizzare i dati, è necessario impostare il tipo di dati su REAL.
Il tipo REAL consente l'utilizzo di valori con 32 bit di lunghezza. Vedere di seguito:-
-3,402823 x 1038 ~ -1,175494 x 10-38, 0,
+1,175494 x 10-38 ~ +3,402823 x 1038

Nome definizione FB	AverageCalc_3Value
Simboli di ingresso	x (tipo REAL), y (tipo REAL), z (tipo REAL)
Simbolo di uscita	score (tipo REAL)
Definizione programma ST	score := (x + y + z) / 3,0;

La sostituzione di un valore con un simbolo viene espressa con " := ".

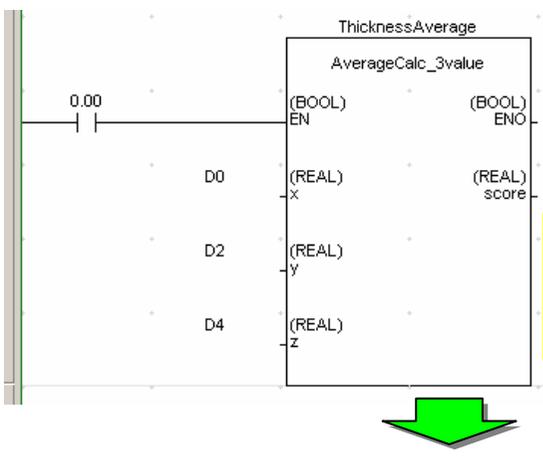
Immettere " ; " (punto e virgola) per completare il codice.

Funzionamento nor
inizione del programma di des
Creazione di una nuova definizione FB
Immissione di variabili
Creazione di un programma ST
Creazione di un programma ladder e verifica
3. Creazione di un blocco funzione con ST
Creare un blocco funzione con il testo strutturato.
Programs
NewProgram1 (00)
Symbols
Section1
END
Insert Function Block
Ladder
Structured Text
From File...
Cut
Copy
Paste
Delete
Allow Docking
Hide
Float In Main Window
Properties
Selezionare l'icona del blocco funzione il cursore del mouse e fare clic con il pulsante destro del mouse.
® Inserisci blocco funzione(I)
® Testo strutturato(S)
NewProject
NewPLC1[CS1G-H] Offline
IO Table
Settings
Memory
Function Blocks
FunctionBlock1
Viene creata una nuova definizione blocco funzione.
Modificare il nome della definizione blocco funzione
AverageCalc_3value
Nota:
non è possibile creare definizioni blocco funzione con nomi che iniziano con "_" (carattere di sottolineatura).
Utilizzare nomi che non iniziano con "_".
Selezionare l'icona di definizione del blocco funzione con il cursore del mouse e fare clic con il pulsante destro del mouse.
Selezionare Incolla.
® Rinomina
Immettere
[AverageCalc_3value]
Tabella variabili
Aprire l'Editor di testo strutturato del blocco funzione
Campo di modifica ST
Selezionare l'icona della definizione blocco funzione con il cursore e fare doppio clic con il pulsante sinistro del mouse.
Internals
Inputs
Outputs
Externals

Funzionamento nor
Immissione di variabili
Creazione di un Programma ST
Creazione di un programma ladder e verifica
4. Immissione di variabili nei blocchi funzione
Selezionare la tabella variabili.
Name
Data Type
AT
Initial Value
Retai...
Comment
EN
BOOL
FALSE
Controls execution of the Func...
Internals
Inputs
Outputs
Externals
Selezionare la scheda Ingressi con il cursore del mouse.
Selezionare Inserisci dal menu a comparsa.
Immettere il nome della variabile
New Variable
Name:
x
Data Type:
REAL
Usage:
Input
Initial Value:
0
Retain
Comment:
Input value
OK
Cancel
Advanced...
Selezionare REAL
Immettere un commento
Immettere i seguenti dati.
Nome
Tipo di dati
Commento
Immettere il simbolo di ingresso x e i simboli di uscita y,z ripetendo la stessa procedura.
Name
Data Type
AT
Initial Value
Retai...
Comment
EN
BOOL
FALSE
Controls execution of the Func...
x
REAL
0.0
Input value 1
y
REAL
0.0
Input value 2
z
REAL
0.0
Input value 3
Variabili di ingresso
ENO
BOOL
FALSE
Indicates successful execution ...
score
REAL
0.0
Avrage
Variabili di uscita
Riferimento: In Intestazione FB è disponibile la funzione di copia e incolla.
Riferimento: L'ordine delle variabili nella tabella FB corrisponde a quello dei parametri nell'istanza FB (istruzione di chiamata) nella normale vista ladder.
Per modificare l'ordine, è possibile trascinare le variabili all'interno della tabella.
ThicknessAverage
AverageCalc_3value
(BOOL) EN
(BOOL) ENO
(REAL) x
(REAL) score
(REAL) y
(REAL) z
Variabili di ingresso
Variabili di uscita
Istanza FB (chiamata)

5. Immissione del programma ST

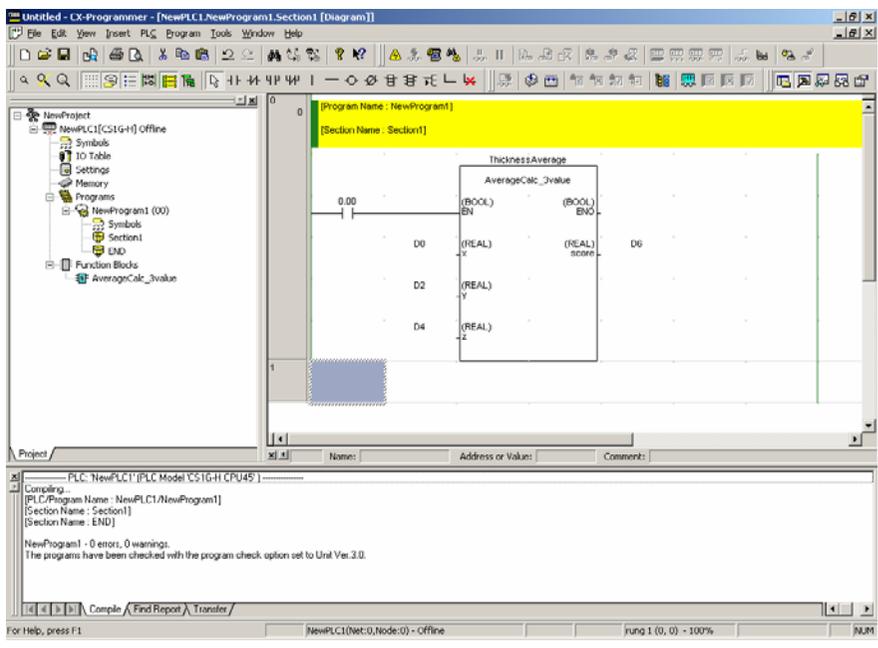
Selezionare il campo dell'editor ST nella finestra Editor di testo strutturato del blocco funzione.

Immettere il testo nel campo: "score := (x + y + z) / 3.0;".

Riferimento: è possibile immettere commenti nel programma ST.
Includere le stringhe dei commenti fra "(*" e "*)", come mostrato di seguito.
Questa soluzione risulta utile per la registrazione dello storico modifiche, delle espressioni del processo e così via.

6. Immissione di FB nel programma ladder e verifica degli errori

Immettere il seguente FB nel programma ladder.
Nome istanza: ThicknessAvarage
Parametri di ingresso: D0, D2, D4
Parametro di uscita: D6

Per passare alla definizione blocco funzione specificata, premere [Maiusc]+[F] quando il cursore si trova nell'istanza blocco funzione.

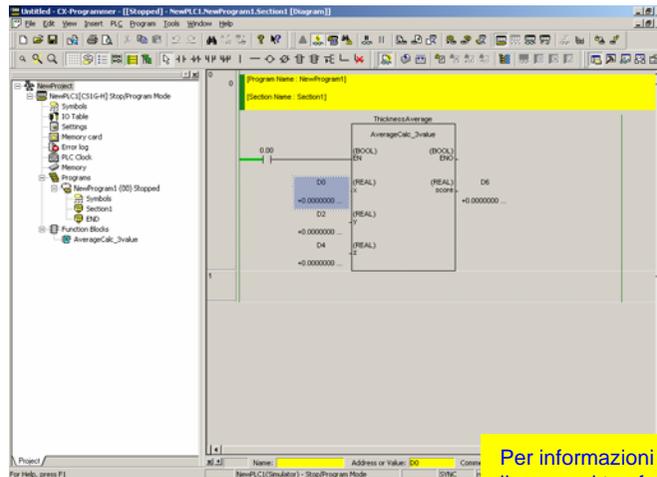
Per informazioni sull'immissione di istanze FB, vedere a pagina 2-7.
La procedura di immissione di istanze FB ST è uguale a quella delle istanze ladder FB.

Prima di trasferire il programma, eseguirne il controllo.

Per informazioni sul controllo dei programmi, vedere a pagina 2-9.
La funzionalità è uguale a quella delle istanze ladder FB.

È possibile modificare o aggiungere variabili al blocco funzione dopo avere immesso l'istanza FB nell'editor ladder. Se si apportano modifiche, il colore della barra bus sinistra del rung contenente il blocco funzione modificato dell'editor ladder cambia.
In questo caso, selezionare l'istanza nell'editor ladder con il cursore del mouse e selezionare Aggiorna istanza blocco funzione (U) dal menu a comparsa.

Trasferisci programma → Monitoraggio

7. Trasferimento di programmi

Attivare la modalità in linea del PLC con CX-Simulator e trasferire il programma.

Per informazioni sull'attivazione della modalità in linea e sul trasferimento del programma, vedere a pagina 2-10.

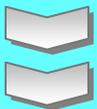
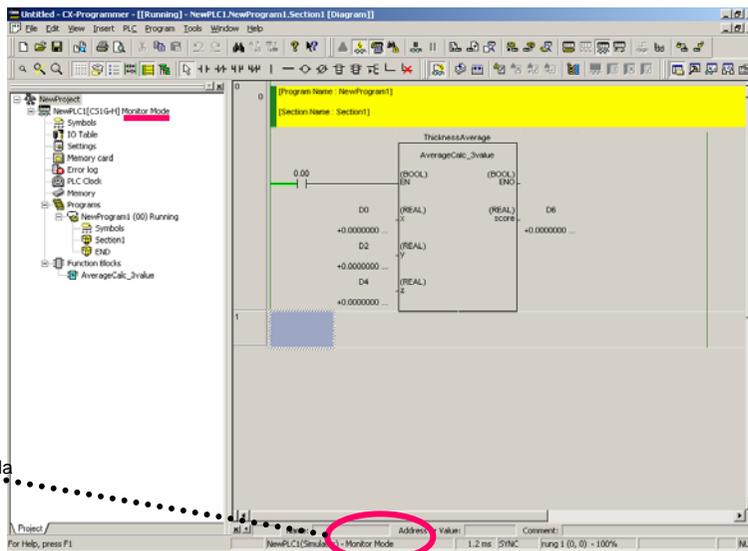
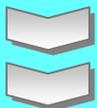
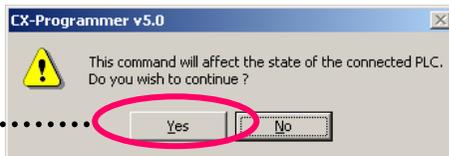
Impostare la modalità Monitor del PLC (Simulatore).

È possibile monitorare lo stato di attivazione/disattivazione dei contatti e delle bobine.

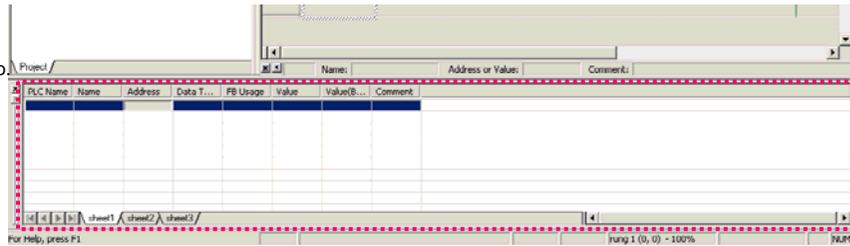
Fare clic su

CX-Programmer v5.0

This command will affect the state of the connected PLC.
Do you wish to continue ?

Yes No

Fare clic su [Sì]

Verificare che nel PLC sia impostata la modalità MONITOR.

8. Monitoraggio dell'esecuzione dei blocchi funzione

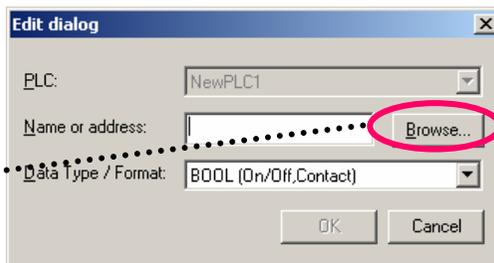
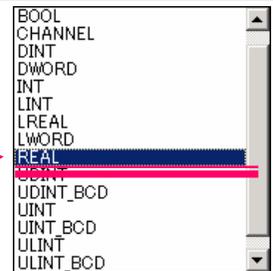
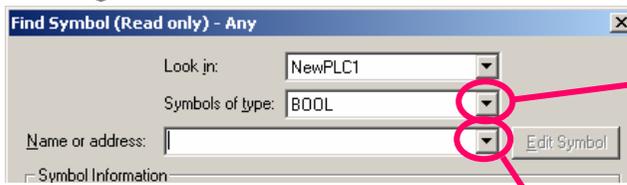
Consente di monitorare il valore corrente dei parametri nell'istanza FB utilizzando la finestra di monitoraggio.

Visualizzare la finestra di monitoraggio.

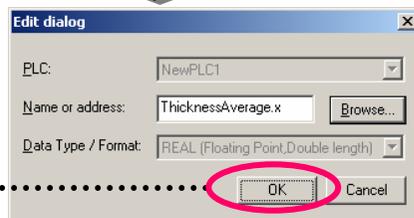
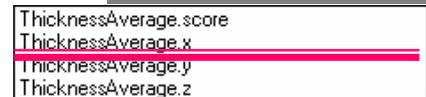
Alt + 3

Aprire la finestra di dialogo Modifica.

ENT

Fare clic sul pulsante Sfoglia.

Fare clic sul pulsante, quindi effettuare le seguenti selezioni:
[Simboli di tipo]
[Nome o indirizzo]

Selezionare REAL (virgola mobile a 32 bit

Selezionare ThicknessAverage.x

Fare clic sul pulsante [OK].

Durante il monitoraggio delle variabili interne nella fase di debug, oltre alla registrazione singola, nella finestra di monitoraggio è possibile eseguire la registrazione collettiva attenendosi alla procedura mostrata qui. Per ulteriori informazioni, consultare la sezione "5-8 Registrazione batch nella finestra di monitoraggio".
Se il blocco funzione è un ladder, è possibile eseguire il monitoraggio. Per ulteriori informazioni, consultare la sezione "5-5 Verifica delle operazioni - 1"

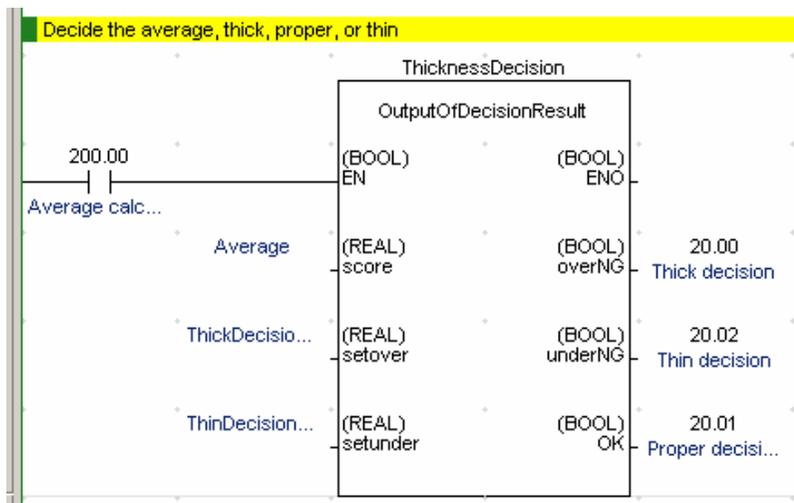
Riferimento: esempio di un programma ST che utilizza IF-THEN-ELSE-END_IF

Il seguente programma ST verifica il valore medio calcolato dall'esempio di pagina 4-7 confrontandolo con un dato intervallo (limite superiore o inferiore).

Definizione FB: OutputOfDecisionResult
Simboli di ingresso: score (tipo REAL), setover (tipo REAL), setunder (tipo REAL)
Simboli di uscita: OK (tipo BOOL), overNG (tipo BOOL), underNG (tipo BOOL)

Programma ST:

```
IF score > setover THEN          (* If score > setover, *)
  underNG := FALSE;              (* Turn off underNG *)
  OK := FALSE;                   (* Turn off OK *)
  overNG := TRUE;                (* Turn on overNG *)

ELSIF score < setunder THEN      (* if score =< setover  and score < setunder then *)
  overNG := FALSE;               (* Turn on overNG *)
  OK := FALSE;                   (* Turn off OK *)
  underNG := TRUE;               (* Turn on underNG *)

ELSE                             (* if setover > score > setunder then*)
  underNG := FALSE;              (* Turn off underNG *)
  overNG := FALSE;               (* Turn off overNG *)
  OK := TRUE;                    (* Turn off OK  *)

END_IF;                          (* end of IF section*)
```

Esempio di un'istanza FB (il nome dell'istanza è "ThicknessDecision")

Capitolo 5
Avanzate
(Creazione di componenti di un programma con FB)

1. Panoramica

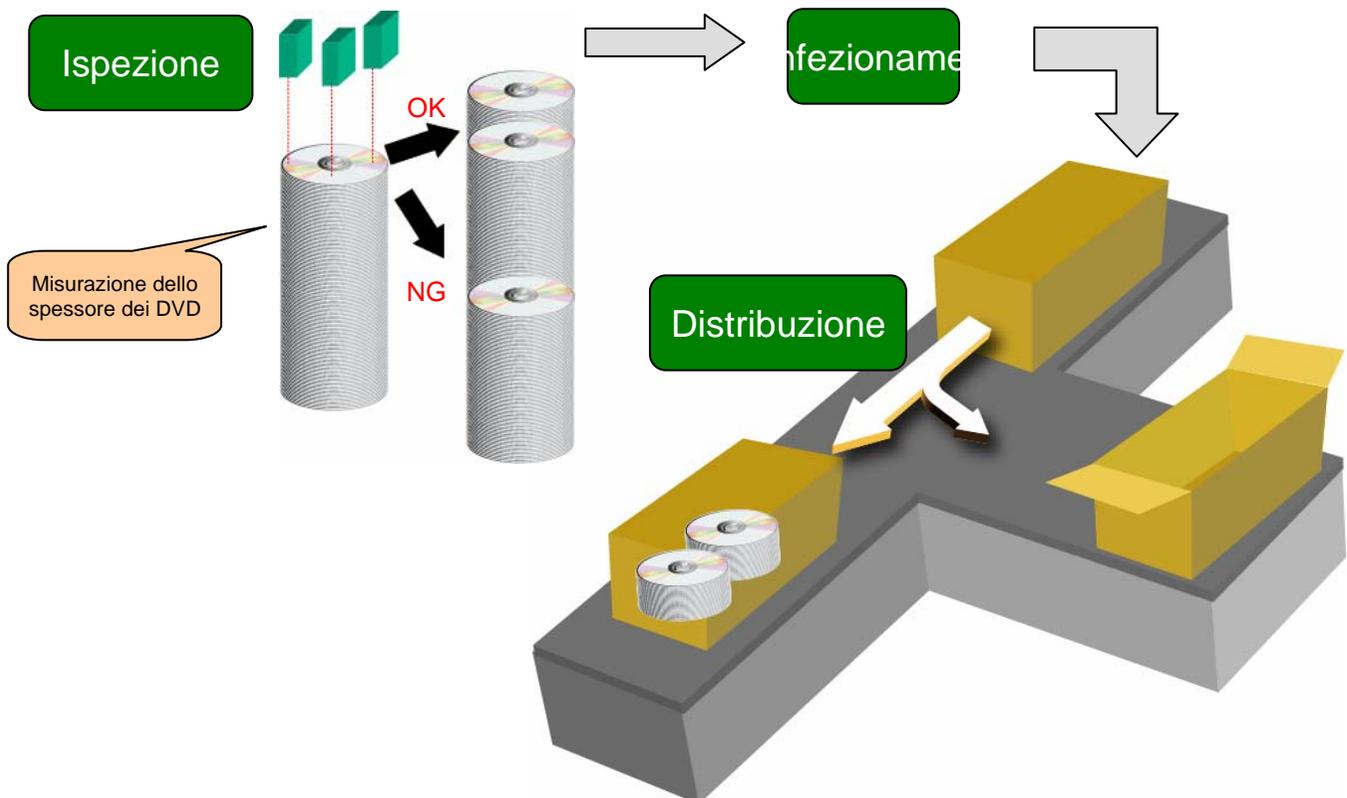
In questo capitolo viene descritto come suddividere in componenti un programma dell'utente con un esempio in cui vengono utilizzati i blocchi funzione.

2. Sviluppo successivo di un programma

Di seguito viene mostrato un flusso di lavoro per la creazione di un programma dell'utente con suddivisione in componenti nel caso dell'applicazione mostrata in basso. È necessario prestare particolare attenzione durante la progettazione del programma.

(1) Progettazione del programma
(2) Creazione di componenti
  (2-1) Immissione del componente FB
  (2-2) Debug del componente FB
  (2-3) Creazione della libreria dei componenti FB (salvataggio del file)
(3) Utilizzo dei componenti nell'applicazione
  (3-1) Importazione dei componenti
  (3-2) Utilizzo dei componenti per il programma
  (3-3) Debug del programma
(4) Avvio

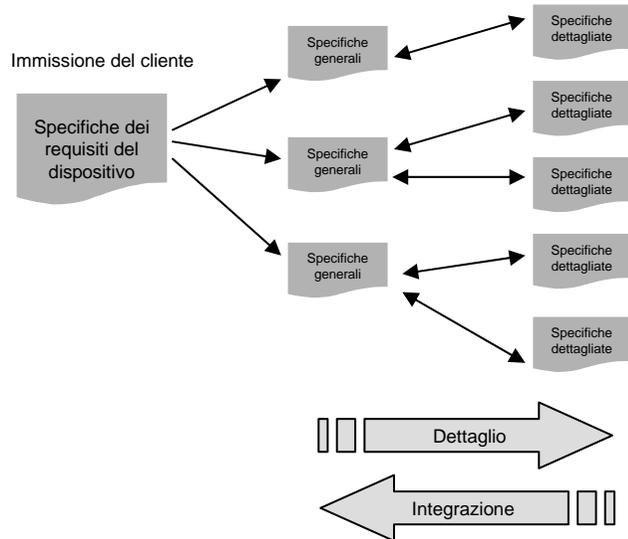
3. Esempio applicativo

Come esempio applicativo, di seguito viene mostrata una macchina per l'ispezione di DVD.
Le fasi fondamentali del processo sono l'ispezione, il confezionamento e la distribuzione.

Ispezione
OK
NG
Misurazione dello spessore dei DVD
nfezioname
Distribuzione

4. Sviluppo successivo di un programma

È possibile creare l'applicazione utilizzando sistemi hardware e programmi software e impostando una serie di requisiti.
Nelle sezioni successive viene descritta la progettazione di un programma utilizzando l'esempio applicativo mostrato in precedenza.

4-1 Panoramica della progettazione

È necessario fornire ripetutamente i dettagli e integrare le specifiche per dividerle e classificarle come mostrato a destra.

4-2 Riepilogo delle specifiche dei requisiti

Di seguito viene fornito un riepilogo delle specifiche dei requisiti dell'applicazione.

Panoramica della macchina per ispezioni di DVD (specifiche dei requisiti)

Req. 1. Il DVD deve essere inserito da un dispositivo di caricamento.

Req. 2. Lo spessore del DVD deve essere misurato in corrispondenza di 3 punti. È necessario calcolare lo spessore medio. Se tale valore rientra nell'intervallo consentito, il DVD deve essere inserito in una torre di immagazzinaggio per i prodotti conformi. In caso contrario, viene inserito in una torre di immagazzinaggio per prodotti difettosi.

Req. 3. I DVD conformi devono essere inseriti nella custodia.

Req. 4. Utilizzare un cartone per i DVD confezionati.

Req. 5. I cartoni devono essere suddivisi in 2 tipi. Prendere in considerazione la frequenza di commutazione per stabilire la durata del finecorsa adiacente all'attuatore del dispositivo di selezione.

Req. 6. Altri requisiti

* Per semplificare la descrizione, in questo documento verrà trattata unicamente una parte del dispositivo (i requisiti sottolineati).

4-3 Dettaglio delle specifiche ed estrapolazione di processi simili

Fornendo i dettagli delle specifiche, sarà possibile individuare processi simili o processi che possono essere utilizzati a livello universale.

Controllo dell'attuatore (esempio di un processo simile)
In questo esempio, il controllo del cilindro per la suddivisione fra prodotti conformi e difettosi e il controllo dell'attuatore per la suddivisione dei cartoni possono essere considerati analoghi. Shown below are extracted requirements for these processes.

- Il processo include 2 attuatori per il movimento bilaterale, che funzionano in base a specifiche condizioni.
- È necessario interbloccare il funzionamento di entrambe le direzioni.
- Per il ripristino del funzionamento, il processo include un segnale di ingresso.

Controllo Average_Threshold (esempio di processo universale)
È necessario estrapolare un processo che verrà utilizzato a livello universale, anche se in questa applicazione verrà utilizzato una sola volta. In questo esempio, viene estrapolato un processo che calcola la media di 3 dati di spessore misurati del DVD e verifica che rientrino nell'intervallo stabilito. Di seguito viene fornito un riepilogo dei requisiti di tale processo.

- È necessario calcolare la media di 3 misurazioni.
- È necessario verificare se il valore medio rientra nell'ambito del limite superiore e inferiore della soglia.

Tali requisiti vengono utilizzati come base per i componenti. I nomi dei componenti vengono definiti come FB "ActuatorContro" FB e FB "AvgValue_ThresholdCheck".

4-3-1 Creazione delle specifiche per i componenti

Il riutilizzo dei componenti può migliorare la produttività dello sviluppo di programmi. Per rendere disponibile il riutilizzo, è importante creare specifiche e inserire commenti per semplificare la comprensione delle specifiche degli ingressi/uscite o del funzionamento senza necessità di osservare il componente.
Si consiglia di descrivere il riferimento alla libreria FB di OMRON.

4-3-2 Esempio della creazione di un componente FB

FB "ActuatorControl"

Deve essere descritto in una sequenza ladder poiché si tratta di un processo di controllo delle sequenze.

[Variabili di ingresso]

Name	Data Type	AT	Initial Value	Retained	Comment
EN	BOOL		FALSE		Controls execution of the Function Block.
PosDirInput	BOOL		FALSE		Input for positive direction
NegDirInput	BOOL		FALSE		Input for negative direction
LSpos	BOOL		FALSE		Limit switch for positive direction
LSneg	BOOL		FALSE		Limit switch for negative direction

[Variabili di uscita]

Name	Data Type	AT	Initial Value	Retained	Comment
ENO	BOOL		FALSE		Indicates successful execution of the Function Block ...
ActuatorPosOut	BOOL		FALSE		Actuator output for positive direction
ActuatorNegOut	BOOL		FALSE		Actuator output for negative direction

[Variabili interne]
Nessuna.

I commenti sulla panoramica del funzionamento e sulle variabili di ingresso e uscita semplificano la comprensione.

```
0
   0  Actuator Control FB
      Summary:
       If Input for positive direction is on, then Actuaor output for positive direction is on until Limit Siwtch for positive direction.
       If Input for negative direction is on, then Actuaor output for negative direction is on until Limit Siwtch for negaive direction.

      Input variable:
      PosDirInput:BOOL
      NegDirInput:BOOL
      LSpos:BOOL
      LSneg:BOOL

      Output variable:
       ActuatorPosOut:BOOL
       ActuatorNegOut:BOOL

      PosDirInput      LSneg      LSpos                 ActuatorPosOut
      ActuatorPosOut
1
   5  NegDirInput      LSpos      LSneg                 ActuatorNegOut
      ActuatorNegOut
```

FB "AvgValue_ThresholdCheck"

Deve essere descritto in ST poiché si tratta di un processo per il confronto e il calcolo numerico.

[Variabili di ingresso]

Name	Data Type	AT	Initial Value	Retained	Comment
EN	BOOL		FALSE		Controls execution of the Function Block.
Input1	REAL		0.0		Input value 1
Input2	REAL		0.0		Input value 2
Input3	REAL		0.0		Input value 3
UpLimit	REAL		0.0		Upper limit value
LowLimit	REAL		0.0		Lower limit value

[Variabili di uscita]

Name	Data Type	AT	Initial Value	Retained	Comment
ENO	BOOL		FALSE		Indicates successful execution of the Function Block...
Result	BOOL		FALSE		OK or NG judge flag

[Variabili interne]

Name	Data Type	AT	Initial Value	Retained	Comment
AvgValue	REAL		0.0		

```
(* Agarage value calculation and check of threshould for three values *)

AvgValue := ( Input1 + Input2 + Input3 ) / 3.0;          (* Divides Input 3 values by 3 *)
IF ((AvgValue <=UpLimit) AND (AvgValue >=LowLimit)) THEN (* Compare the agarage value if below of upper limit or above of lower limit *)
    Result := TRUE;
    ELSE
    Result := FALSE;
END_IF;
```

Nota: si consiglia per quanto possibile di attribuire nomi generici a FB e alle variabili nel diagramma ladder e in ST e di evitare nomi specifici relativi alla funzione al momento della creazione.

4-4 Integrazione degli FB

A questo punto sono stati estrapolati i componenti dettagliati del processo. I componenti dell'applicazioni verranno creati combinandoli nelle sezioni successive.

4-4-1. Combinazione di componenti esistenti - DVD_ThickSelectControl

Il Req. 2. "Lo spessore del DVD deve essere misurato in corrispondenza di 3 punti. È necessario calcolare lo spessore medio. Se tale valore rientra nell'intervallo consentito, il DVD deve essere inserito in una torre di immagazzinaggio per i prodotti conformi. In caso contrario, viene inserito in una torre di immagazzinaggio per prodotti difettosi" può essere considerato come un processo che combina "AvgValue_ThresholdCheck" e "ActuatorControl", descritti nella sezione precedente. La "combinazione" di tali componenti consente di creare il componente integrato FB "DVD_ThickSelectControl". Di seguito viene mostrato un esempio del blocco funzione da creare.

[Variabili di ingresso]

Name	Data Type	AT	Initial Value	Retained	Comment
EN	BOOL		FALSE		Controls execution of the Function Block.
LSright	BOOL		FALSE		Limit switch for cylinder right direction
LSleft	BOOL		FALSE		Limit switch for cylinder left direction
Measure1	REAL		0.0		Measurement result 1 of DVD thickness (mm)
Measure2	REAL		0.0		Measurement result 2 of DVD thickness (mm)
Measure3	REAL		0.0		Measurement result 3 of DVD thickness (mm)

[Variabili di uscita]

Name	Data Type	AT	Initial Value	Retained	Comment
ENO	BOOL		FALSE		Indicates successful execution of the Function Block ...
CylinderRightOn	BOOL		FALSE		Output for sylinder right direction
CylinderLeftOn	BOOL		FALSE		Output for sylinder left direction

[Variabili interne]

Name	Data Type	AT	Initial Value	Retained	Com
WorkMove	FB [ActuatorControl]				
DVDThickJudge	FB [AvgValue_ThresholdCheck]				
Judge	BOOL		FALSE		
_Judge	BOOL		FALSE		

Il nome e i nomi delle variabili interne di questo FB includono "DVD" o "Cylinder" perché è stato creato specificamente per l'applicazione.

È possibile chiamare un blocco funzione dall'interno di un altro blocco funzione. Tale operazione viene definita "nidificazione".

Per effettuare la nidificazione, è necessario dichiarare una variabile di tipo FUNCTION BLOCK(FB) come variabile interna per utilizzare il nome della variabile come istanza.

4-4-2. Aggiunta di funzioni ai componenti esistenti - WorkMoveControl_LSONcount

Il Req. 5. "I cartoni devono essere suddivisi in 2 tipi. Prendere in considerazione la frequenza di commutazione per stabilire la durata del finecorsa adiacente all'attuatore del dispositivo di selezione" può essere implementato contando il numero di attivazioni e disattivazioni di un finecorsa come ingresso di "ActuatorControl". Questo componente viene definito FB "WorkMoveControl_LSONcount". Di seguito viene mostrato un esempio del blocco funzione da creare.

[Variabili di ingresso]

Name	Data Type	AT	Initial Value	Retained	Comment
EN	BOOL		FALSE		Controls execution of the Function Block.
RightDirInput	BOOL		FALSE		Condition to move actuator to right direction
LeftDirInput	BOOL		FALSE		Condition to move actuator to left direction
LSright	BOOL		FALSE		Limit switch for acutuator right direction
LSleft	BOOL		FALSE		Limit switch for acutuator left direction
Reset	BOOL		FALSE		Resets number of times for opening - closing li...

[Variabili di uscita]

Name	Data Type	AT	Initial Value	Retained	Comment
ENO	BOOL		FALSE		Indicates successful execution of the Functio...
ActuatorRightOn	BOOL		FALSE		Output for actuator right direction
ActuatorLeftOn	BOOL		FALSE		Output for actuator left direction
LS_ONnumber	LINT		0		

[Variabili interne]

Name	Data Type	AT	Initial Value	Retained	Comment
PrevCycleLS	BOOL		FALSE		
WorkMove	FB [ActuatorControl]				

```
(* Work move control and count of number of times open - close of limit switch *)
(* Created by: machine development div. Yamada: 10-01-2005 *)

(* Resets number of times opening - closing limit siwtch *)
IF Reset = TRUE THEN
    PrevCycleLS := FALSE;
END_IF;


(* Calls WorkMove (instance of ActuatorControl FB) *)
WorkMove(RightDirInput, LeftDirInput, LSright, LSleft, ActuatorRightOn, ActuatorLeftOn);

(* Counts number of times opening - closing limit switch *)
IF PrevCycleLS = FALSE and LSright = TRUE THEN
    LS_ONnumber := LS_ONnumber+1;
END_IF;
PrevCycleLS := LSright;  (* Copies LSright to compare at next
```

Il blocco funzione ladder viene richiamato da ST.

Come richiamare il blocco funzione da ST

FB da richiamare: MyFB

Istanza di MyFB dichiarata in ST: MyInstance

Variabile I/O del blocco funzione da richiamare:
- Ingresso: Input1, Input2
- Uscita: Output1, Output2

Variabile I/O da trasferire al blocco funzione in ST:
- Ingresso: STInput1, STInput2
- Uscita: STOutput1, STOutput2

In questo esempio, la chiamata dell'istanza FB da ST deve essere descritta come

MyInstance(Input1 := STInput1, Input2 := STInput2, Output1 => STOutput1, Output2 => STOutput2);

Una volta descritte tutte le variabili di ingresso/uscita, è possibile omettere la descrizione delle variabili e gli operatori di assegnazione in quella da richiamare.

MyInstance(STInput1, STInput2, STOutput1, STOutput2);

Descrivendo le variabili e gli operatori di assegnazione nell'istanza da richiamare, è possibile descrivere solo una parte delle variabili di ingresso/uscita.

MyInstance(Input1 := STInput1, Output2 => STOutput2);

4-5. Descrizione del programma generale

Perché i componenti (FB) qui analizzati possano funzionare come programma, è necessario creare un circuito che richiami un componente integrato dal programma ladder principale.
* L'esempio fornito fa riferimento unicamente al Req. 2 e al Req. 5.

[Variabili globali]

Name	Data Type	Address / Value	Rack Location	Usage
StageA_BoxSelect	FB [WorkMoveControl_LSONcount]	N/A [Auto]		
StageA_DVDThickSelect	FB [DVD_ThickSelectControl]	N/A [Auto]		

* Le variabili di istanze diverse da quelle utilizzate per il blocco funzione vengono omesse.

StageA_DVDThickSelect
DVD_ThickSelectControl
0.00 — (BOOL) EN — (BOOL) ENO
1.00 — (BOOL) LSright — (BOOL) CylinderRightOn
1.01 — (BOOL) LSleft — (BOOL) CylinderLeftOn
D0 — (REAL) Measure1
D2 — (REAL) Measure2
D4 — (REAL) Measure3

L'uscita ON del cilindro destro (2.00) o del cilindro sinistro (2.01) è attivabile utilizzando un finecorsa (1.00, 1.01) in corrispondenza dell'unità cilindro per distribuire i DVD e le tre misurazioni (D0-D5) dello spessore dei DVD come ingressi e impostando su ON il contatto 0.00.

StageA_BoxSelect
WorkMoveControl_LSONcount
P_On — (BOOL) EN — (BOOL) ENO
W0.00 — (BOOL) RightDirInput — (BOOL) ActuatorRightOn
W0.01 — (BOOL) LeftDirInput — (BOOL) ActuatorLeftOn
3.00 — (BOOL) LSright — (LINT) LS_ONnumber
3.01 — (BOOL) LSleft
0.10 — (BOOL) Reset

L'attuatore destro (4.00) o quello sinistro (4.01) vengono impostati su ON in base alle condizioni di un finecorsa (3.00, 3.01) in corrispondenza dell'estremità dell'attuatore se vengono immessi l'ingresso dell'operazione (W0.00) e l'ingresso dello spostamento a sinistra (W0.01) per spostare un cartone contenente i DVD verso destra o verso sinistra.
Inoltre, a D10-13 viene fornito il conteggio delle commutazioni del finecorsa.

Perché il nome dell'istanza è "StageA*"?**
Benché non sia esplicitamente descritto nell'esempio applicativo, è possibile creare un programma per la fase B appena aggiunta solo descrivendo un'istanza "StageB***" nel programma e impostando i necessari parametri, senza registrare un nuovo blocco funzione.
In base alle caratteristiche stabilite da Omron, un blocco funzione può includere più istanze. Utilizzando una definizione di blocco funzione con verifica delle operazioni (algoritmo), è possibile creare un programma solo assegnandone l'indirizzo.

4-5-1. Struttura del programma generale

In questa sezione viene verificata la struttura del programma generale inclusi i componenti (blocchi funzione) creati.

[Programma principale]

StageA_DVDThickSelect / DVD_ThickSelectControl: 0.00 EN → ENO; 1.00 LSright → CylinderRightOn 2.00; 1.01 LSleft → CylinderLeftOn 2.01; D0 Measure1; D2 Measure2; D4 Measure3

StageA_BoxSelect / WorkMoveControl_LSONcount: P_On EN → ENO; W0.00 RightDirInput → ActuatorRightOn 4.00; W0.01 LeftDirInput → ActuatorLeftOn 4.01; 3.00 LSright → LS_ONnumber D10; 3.01 LSleft; 0.10 Reset

The upper limit is 1.26mm, the lower limit is 1.14mm. (1.20mm (+-5%))

DVDThickJudge / AvgValue_ThresholdCheck: EN → ENO; Measure1 Input1; Measure2 Input2; Measure3 Input3 → Result; +1.26 UpLimit; +1.14 LowLimit

Judge → _Judge

WorkMove / ActuatorControl: EN → ENO; Judge PosDirInput → ActuatorPosOut CylinderRightOn; _Judge NegDirInput → ActuatorNegOut CylinderLeftOn; LSright LSpos; LSleft LSneg

Actuator Control FB
Summary:
If input for positive direction is on, then Actuator output for positive direction is on until Limit Switch for positive direction.
If input for negative direction is on, then Actuator output for negative direction is on until Limit Switch for negative direction.

Input variable:
PosDirInput BOOL
NegDirInput BOOL
LSpos BOOL
LSneg BOOL

Output variable:
ActuatorPosOut BOOL
ActuatorNegOut BOOL

```
(* Work move control and count of number of times open - close of limit switch *)
(* Created by: machine development div. Yamada: 10-01-2005 *)

(* Resets number of times opening - closing limit siwtch *)
IF Reset = TRUE THEN
    PrevCycleLS := FALSE;
END_IF;

(* Calls WorkMove (instance of ActuatorControl FB) *)
WorkMove(RightDirInput, LeftDirInput, LSright, LSleft, ActuatorRightOn, ActuatorLeftOn);

(* Counts number of times opening - closing limit switch *)
IF PrevCycleLS = FALSE and LSright = TRUE THEN
    LS_ONnumber := LS_ONnumber+1;
END_IF;
PrevCycleLS := LSright;  (* Copies LSright to compare at next execution *)
```

I nomi delle istanze e i nomi FB possono essere descritti come segue: (il nome FB viene mostrato fra [ ])

In un programma strutturato, in particolare per modificare un componente (FB) di livello inferiore, è importante comprendere i rapporti principale/secondario e la condivisione dei componenti nel caso in cui sia necessario cancellare il flusso del processo a causa di debug e così via.
Come documentazione del progetto, si consiglia di creare un diagramma comprensibile della struttura del programma generale.

Per informazioni sulla struttura software creata dagli FB, CX-Programmer Ver. 6.0 include "Visualizzatore istanza FB", attivabile premendo [Alt]+[5]. È inoltre possibile verificare che l'indirizzo sia stato assegnato all'istanza FB.

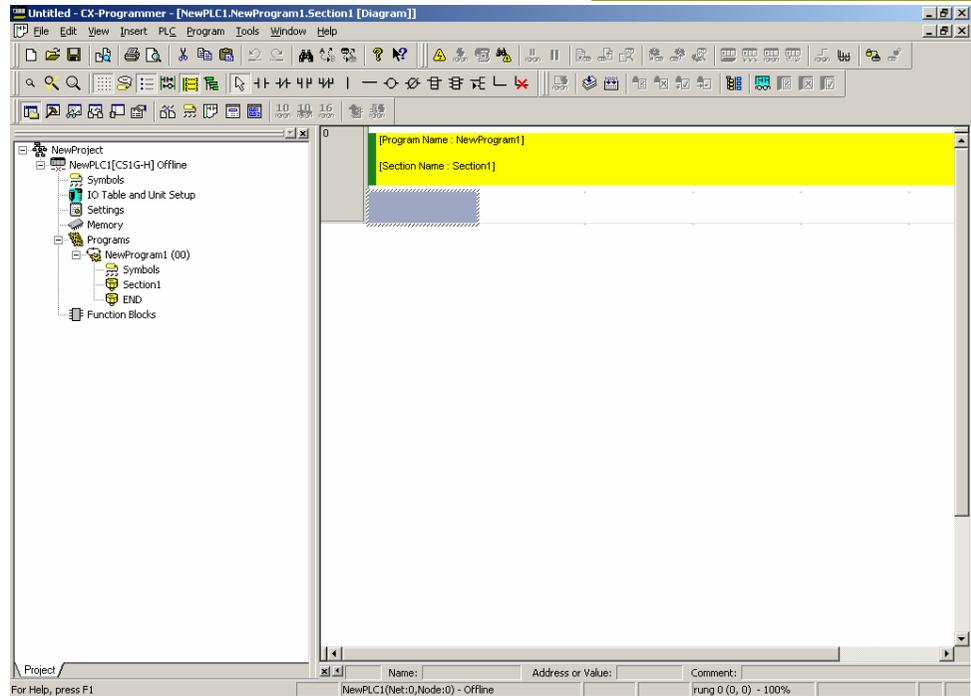
Name	Data Type	Address	Comment
EN	BOOL	H513.00	Controls execution of the Function Block.
Input1	REAL	H514	Input value 1
Input2	REAL	H516	Input value 2
Input3	REAL	H518	Input value 3
LowLimit	REAL	H522	Lower limit value
UpLimit	REAL	H520	Upper limit value

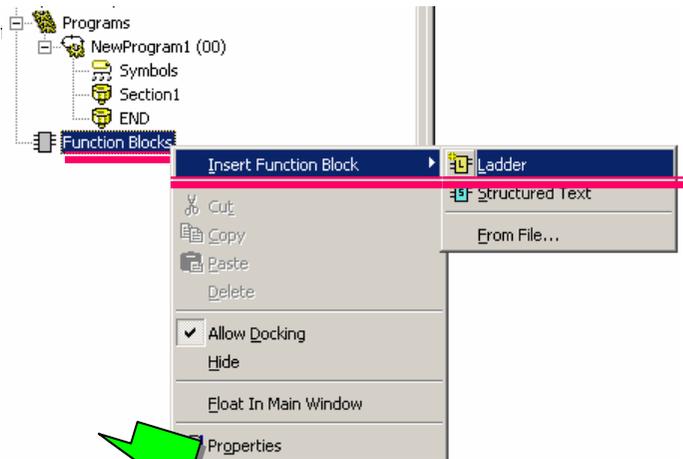
5. Immissione della definizione FB

In questa sezione viene descritta l'immissione di un programma progettato e del relativo debug.
È necessario creare un nuovo progetto e immettere il componente FB "ActuatorControl" mostrato a pagina 5-4.

5-1. Creazione di un nuovo progetto e impostazione del modello PLC/tipo CPU

Vedere pagina 2-3 e creare un nuovo progetto.

! Selezionare un modello di PC fra i seguenti per utilizzare i blocchi funzione.
CS1G-H, CS1H-H, CJ1G-H, CJ1H-H, CJ1M

5-2. Creazione di un blocco funzione delle definizioni ladder

Creare il blocco funzione delle de

Spostare il cursore del mouse su un blocco funzione e clic con il pulsante destro del mouse.
Selezionare
→ Ινσερισχι βλοχχο φυνζιονε
→ Λαδδερ

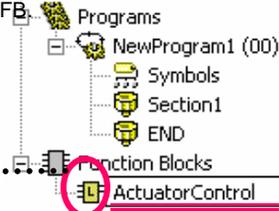
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
Insert Function Block
Ladder
Structured Text
From File...
Cut
Copy
Paste
Delete
Allow Docking
Hide
Float In Main Window
Properties

A questo punto il blocco funzione è stato creato.

Function Blocks
FunctionBlock1

5-3. Immissione di un programma ladder FB

Modificare il nome della definizione FB.

Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
ActuatorControl

Attenzione:
non è possibile creare un nome di definizione blocco funzione che inizia con "_".
Il carattere iniziale del nome deve essere diverso da "_".

Spostare il cursore del mouse sull'icona di un blocco funzione copiato , quindi fare clic con il pulsante destro del mouse.
Selezionare
® Rinomina
Immettere [ActuatorControl].

Aprire l'editor ladder FB.

Spostare il cursore del mouse sull'icona di un blocco funzione , quindi fare doppio clic per aprire l'editor di testo strutturato del blocco funzione.

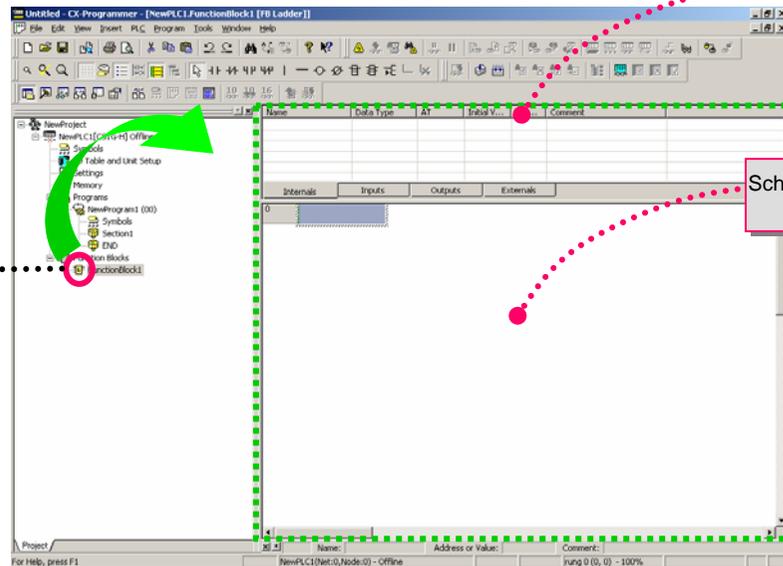



Tabella variabili

Schermata di immissione del ladder

Selezionare la tabella delle variabili e registrare le variabili nel blocco funzione.
È necessario registrare tutte le variabili del blocco funzione "A

Nota: l'ordine delle variabili deve essere uguale a quello dell'istanza FB.
Per modificare l'ordine delle variabili, selezionare e trascinare il nome della variabile.

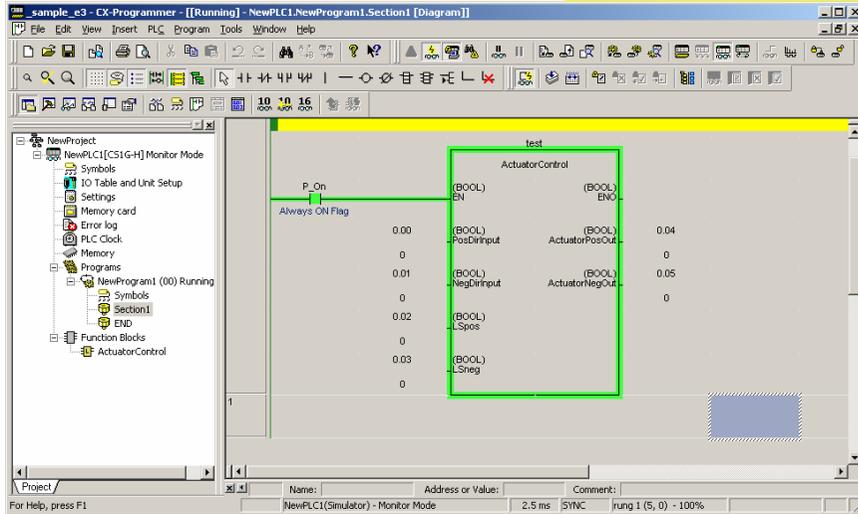
Selezionare la schermata di immissione del ladder, quindi immettere un programma ladder.
È necessario registrare tutte le variabili del blocco funzione "ActuatorControl" di pagina 5-4.

Nota: Benché sia possibile immettere un circuito nell'editor ladder FB con una procedura analoga all'editor ladder principale, non è consentito immettere indirizzi nel blocco funzione.

Nota: per immettere l'elenco delle variabili in un commento, è possibile selezionare e copiare una variabile dalla relativa tabella. Tale opzione risulta utile per ottimizzare l'immissione.

5-4. Trasferimento del programma

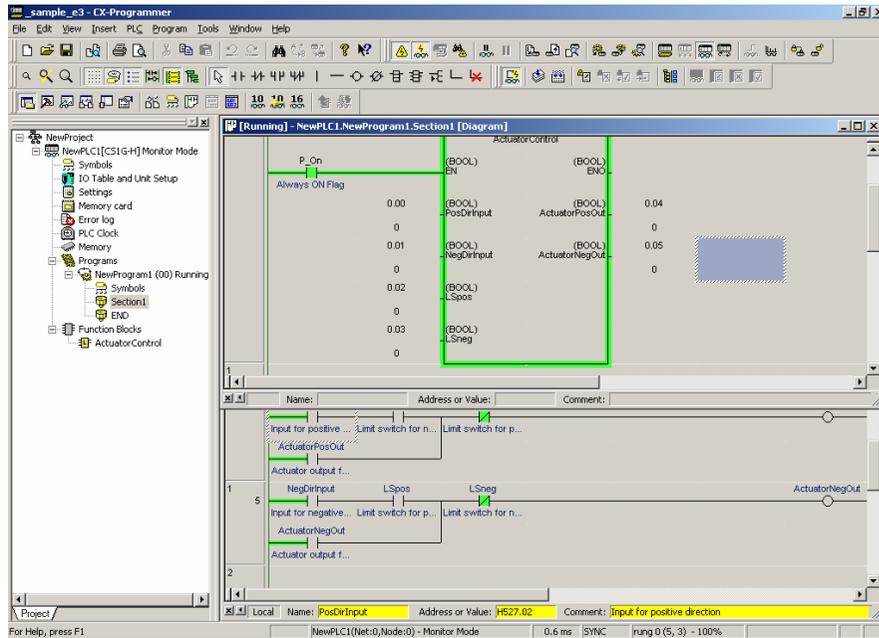
Effettuare una connessione in linea a CX-Simulator, trasferire un programma, quindi impostare la modalità Monitor del PLC (simulatore).

Per informazioni sulle connessioni in linea e il trasferimento di un programma, vedere a pagina 2-10.

5-5. Verifica delle operazioni-1

Modificare il valore corrente del parametro dell'istruzione di chiamata FB nel ladder principale, quindi verificare il funzionamento del blocco funzione "ActuatorControl".
Monitorare innanzitutto l'istanza del blocco funzione ActuatorControl.

Spostare il cursore sull'istruzione di chiamata FB, quindi fare doppio clic oppure fare clic sul pulsante .

Viene monitorata l'istanza ladder FB (in base all'indirizzo assegnato).

Visualizzare contemporaneamente il ladder principale e l'istanza FB (il ladder FB richiamato dal ladder principale), quindi verificare l'operazione modificando il valore corrente del parametro dell'istruzione di chiamata FB nel ladder principale.

5-6. Verifica delle operazioni-2

Immettere i seguenti valori di parametri dell'istruzione di chiamata FB e verificare se è necessario fornire il risultato previsto. In questo esempio viene mostrato solo (1), ma è necessario verificare tutte le combinazioni delle condizioni.

(1) Stato iniziale: Attivare 0.03. => 0.04 e 0.05 devono essere disattivati. Lo stato della schermata di monitoraggio del ladder dell'istanza FB deve corrispondere al valore.

(2) Funzionamento attuatore in direzione avanti-1: Attivare 0.00 => 0.04 deve essere attivato. Lo stato della schermata di monitoraggio del ladder dell'istanza FB deve corrispondere al valore.

(3) Funzionamento attuatore in direzione avanti-2: Disattivare 0.03 => 0.04 deve essere attivato e 0.05 deve essere disattivato. Lo stato della schermata di monitoraggio del ladder dell'istanza FB deve corrispondere al valore.

(4) Funzionamento attuatore in direzione avanti-3: Attivare 0.03 => 0.04 deve essere disattivato e 0.05 deve essere disattivato. Lo stato della schermata di monitoraggio del ladder dell'istanza FB deve corrispondere al valore.

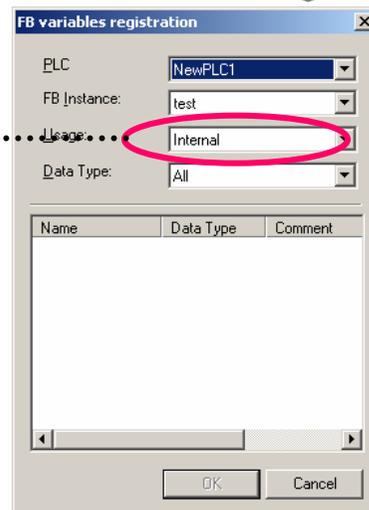
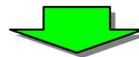
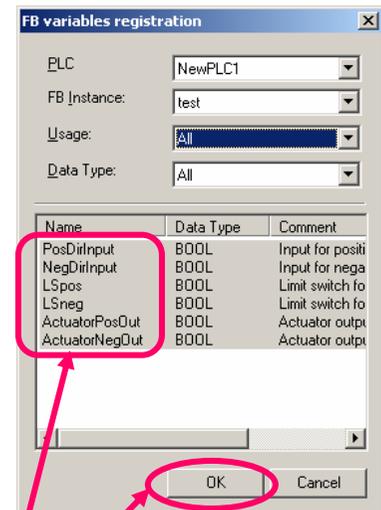
Spostare il cursore su 0.03 e premere il tasto [ENT].

5-7. Immissione/debug di altre definizioni FB

Fino a questo punto sono state descritte le procedure di immissione e debug del blocco funzione "ActuatorControl". È necessario immettere ed eseguire il debug di altre definizioni FB.

5-8. Registrazione batch nella finestra di monitoraggio

Per il debug, è possibile utilizzare la registrazione batch dell'indirizzo dell'istanza FB nella finestra di monitoraggio anziché il monitor ladder FB.

Spostare il cursore sull'istruzione FB da registrare, fare clic con il pulsante destro del mouse e selezionare [Registra in finestra di monitoraggio] dal menu.

Se necessario selezionare Utilizzo e Tipo di dati.

Selezionare il nome da registrare, quindi premere il pulsante [OK].

5-9. Esecuzione di step con la funzione di simulazione

Impostando il punto di interruzione della funzione di simulazione e utilizzando la funzione Esecuzione step, è possibile interrompere l'esecuzione del programma e verificare con facilità lo stato di elaborazione durante l'esecuzione del programma.

Questa funzione può essere utilizzata con CX-One Ver. 1.1 e successive (CX-Programmer Ver. 6.1, CX-Simulator Ver. 1.6 e successive)

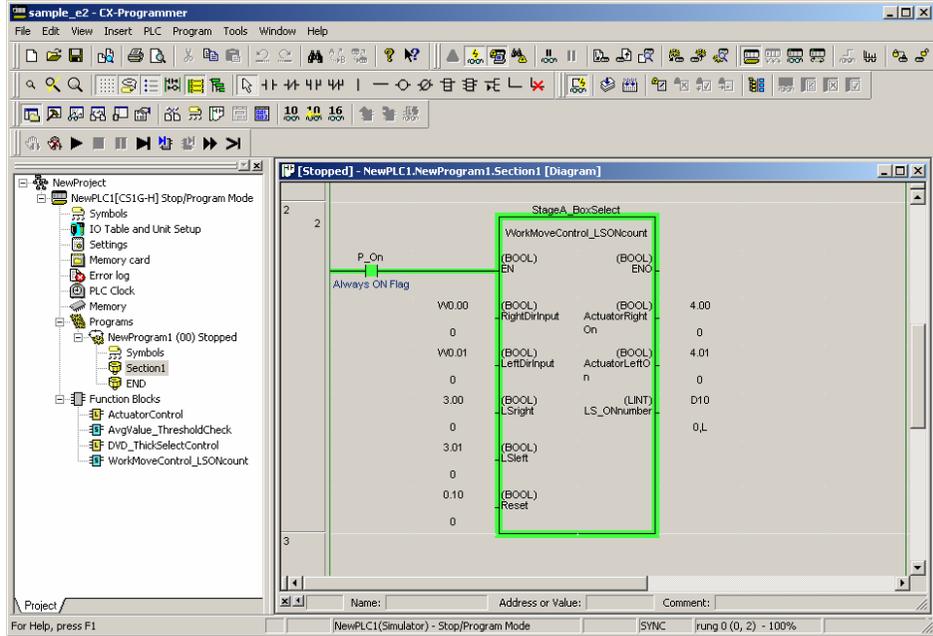
5-9-1. Informazioni sul pulsanti di simulazione

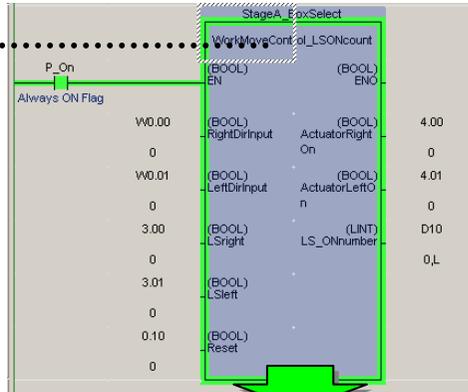
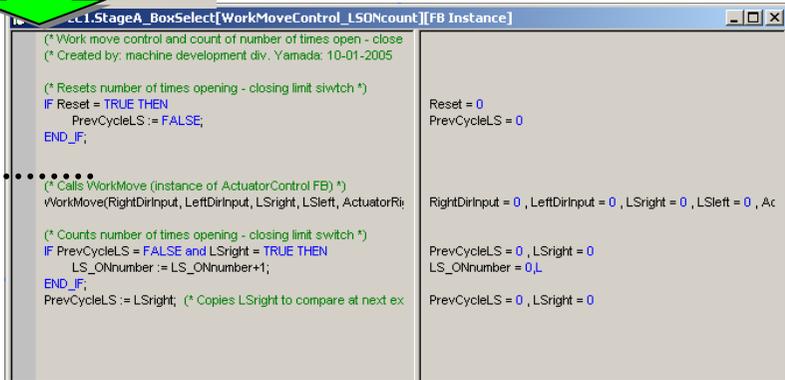
Di seguito vengono illustrati i pulsanti della barra degli strumenti utilizzati con la funzione di simulazione. Viene illustrata la funzione di ciascun pulsante.

Pulsanti di simulazione

	Imposta/Cancella punto di interruzione (tasto F9)	Selezionare le posizioni (ladder, ST) dove effettuare l'interruzione durante l'esecuzione della simulazione. Premendo questo pulsante verrà visualizzato un segno rosso.
	Cancella tutti i punti di interruzione	Elimina un punto di interruzione (segno rosso) impostato con il pulsante Imposta punto di interruzione.
	Esegui (modalità Monitor) (tasto F8)	Consente di eseguire il programma utente e di passare dalla modalità Run a quella Monitor.
	Stop (modalità Program)	Consente di interrompere l'esecuzione del programma e di passare dalla modalità Run a quella Program.
	Pausa	L'esecuzione del programma viene sospesa in corrispondenza del cursore.
	Esecuzione step (tasto F10)	Consente di eseguire uno step del programma utente. Nel caso di un ladder, lo step è costituito da un'istruzione, mentre nel caso del testo strutturato è costituito da una riga.
	Esegui step (tasto F11)	Consente di eseguire uno step del programma utente. Nei casi in cui la posizione del cursore richiami l'istruzione di chiamata FB, effettua il trasferimento all'istanza FB chiamata (ladder o ST).
	Esci da step (Maiusc+tasto F11)	Consente di eseguire uno step del programma utente. Nei casi in cui la posizione del cursore corrisponda all'istanza FB, effettua il trasferimento all'istruzione di chiamata FB di base.
	Esecuzione step continua	Consente di eseguire uno step del programma, ma esegue automaticamente e in modo continuo gli step dopo avere effettuato una pausa per un determinato periodo di tempo.
	Esecuzione a scansione	Consente di eseguire la scansione di un programma (un ciclo).

5-9-2. Impostazione del punto di interruzione ed esecuzione degli step

Di seguito viene fornita una spiegazione in cui viene utilizzata come esempio la funzione di simulazione Debug FB "WorkMoveControl_LSONcount".

Passare dalla modalità Run a quella Monitor.
Visualizzare l'istanza FB "WorkMoveControl_LSONcount".

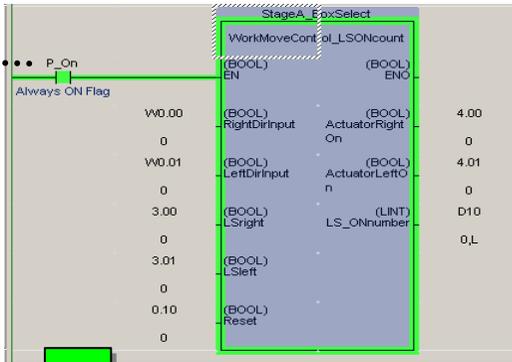
Spostare il cursore all'interno dell'istruzione di chiamata FB, quindi fare doppio clic oppure fare clic sul pulsante .

I valori correnti delle variabili corrispondenti al programma sono monitorati nell'istanza ST FB (con l'indirizzo assegnato).

Impostare il valore corrente nel parametro dell'istruzione di chiamata FB e confermare la condizione di esecuzione.

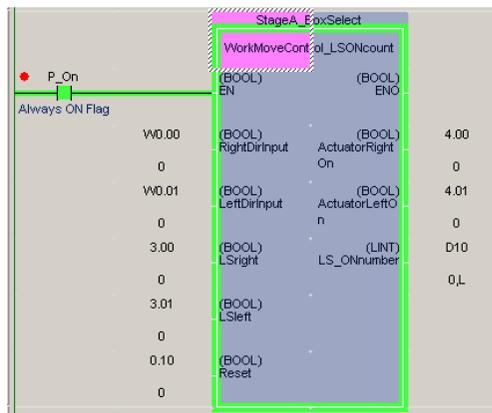
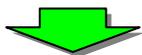
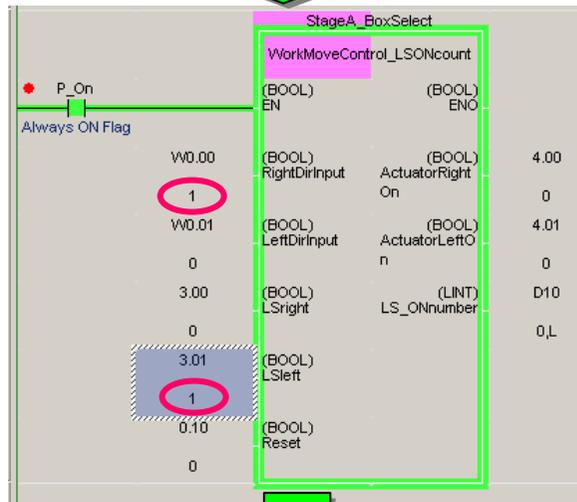
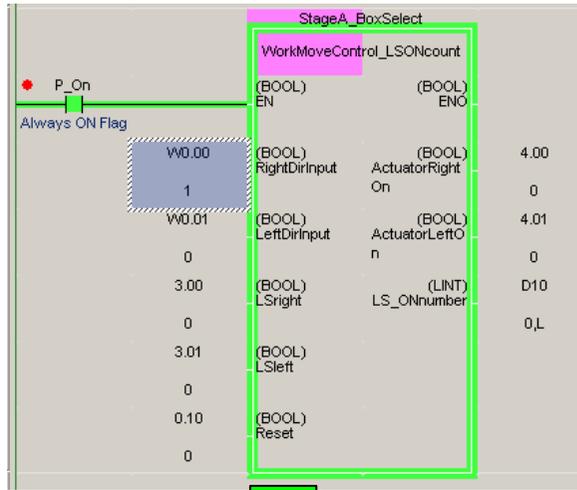
Impostare i seguenti casi:

- RightDirInput: ON
- LeftDirInput: OFF
- LSright: OFF
- LSleft: ON
- Reset: OFF

In questo caso, si prevedono le seguenti uscite:

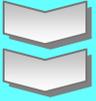
- ActuatorRightOn: ON
- ActuatorLeftOn: OFF
- LS_ONnumber: 1

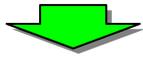
Spostare il cursore sull'ingresso di sinistra dell'istruzione di chiamata FB, quindi fare clic sul pulsante

Il programma si interrompe in corrispondenza del punto di interruzione.

Fare clic sul pulsante .

Eseguire il contatto di ingresso del punto di interruzione. L'interruzione ha luogo in corrispondenza dello step successivo dell'istruzione di chiamata FB.

Impostare su ON il parametro di ingresso "RightDirInput" e "LSleft" nell'istruzione di chiamata ST.

A questo punto sono stati impostati i parametri di ingresso necessari.

Fare clic s… sante .

Posizione dell'esecuzione di Monitor ST

Il cursore si sposta sulla prima riga del programma ST richiamato.

Fare due vc ic sul pulsante .

```
(* Calls WorkMove (instance of ActuatorControl FB) *)
WorkMove(RightDirInput, LeftDirInput, LSright, LSleft, ActuatorRi
ut = 0 , LSright = 1 , LSleft = 0 , ActuatorRightOn = 1 , Actuato
```

Transizioni dal programma ST al programma ladder FB richiamato.

NewPLC1.StageA_BoxSelect.WorkMove[ActuatorControl][FB Instance]

ActuatorNegOut BOOL

PosDirInput LSneg LSpos ActuatorPosOut
Input for positi... Limit switch fo... Limit switch fo...
ActuatorPosOut
Actuator outp...
NegDirInput LSpos LSneg ActuatorNegOut
Input for negat... Limit switch fo... Limit switch fo...
ActuatorNegOut
Actuator outp...

Confermare la correttezza delle condizioni di ingresso dopo il trasferimento dal programma ST al parametro richiamato.

Local Name: PosDirInput Address or Value: H526.02 Comment: Input for positive direction

Fare cinqu e clic sul pulsante .

NewPLC1.StageA_BoxSelect.WorkMove[ActuatorControl][FB Instance]

ActuatorNegOut BOOL

PosDirInput LSneg LSpos ActuatorPosOut
Input for positi... Limit switch fo... Limit switch fo...
ActuatorPosOut
Actuator outp...
NegDirInput LSpos LSneg ActuatorNegOut
Input for negat... Limit switch fo... Limit switch fo...
ActuatorNegOut
Actuator outp...

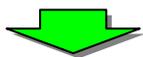
Verificare che il valore dell'uscita "ActuatorPosOut" sia quello previsto.

Local Name: NegDirInput Address or Value: H526.03 Comment: Input for negative direction

Fare clic su sante .

La conferma è stata completata.
Tornare al programma ST di chiamata.

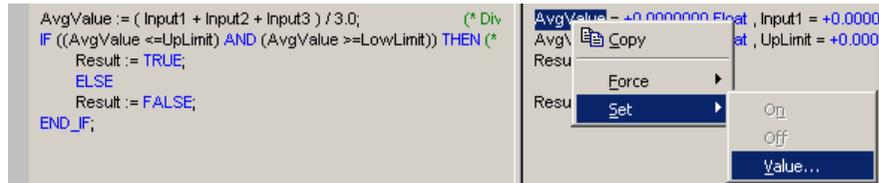
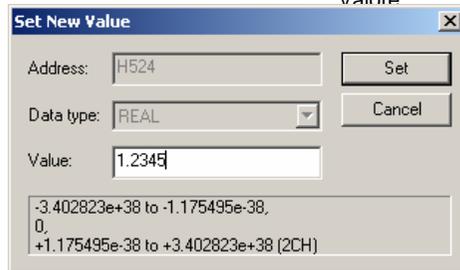
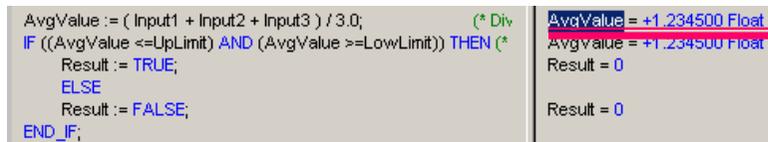
```
(* Calls WorkMove (instance of ActuatorControl FB) *)
WorkMove(RightDirInput, LeftDirInput, LSright, LSleft, ActuatorRi
ut = 0 , LSright = 1 , LSleft = 0 , ActuatorRightOn = 1 , Actuato
```

Verificare che il risultato dell'elaborazione del circuito precedente sia rispecchiato correttamente nella schermata di monitoraggio del programma ST di chiamata.

Fare clic su ante .

Effettuare il trasferimento al programma ladder di chiamata.

Verificare che il parametro di uscita sia correttamente rispecchiato.

Suggerimento

Per modificare del valore del parametro corrente con il programma ST, attenersi alla seguente procedura.

Selezionare il parametro da modificare con il cursore del mouse, fare clic con il pulsante destro del mouse e selezionare Imposta Þ Valore.

Impostare il valore e fare clic sul pulsante [Imposta].

6. Creazione della libreria delle definizioni FB

Per riutilizzare la definizione FB verificata con le operazioni, è necessario incorporarla nella libreria (file). Controllare la gerarchia utilizzando l'area di lavoro del progetto e Visualizzatore istanza FB, quindi determinare la definizione FB da incorporare nella libreria. In questo caso, si tratta del blocco funzione "DVD_ThickSelectControl".

Selezionare il blocco funzione "DVD_ThickSelectControl", fare clic con il pulsante destro del mouse e selezionare [Salva blocco funzione su file] dal menu visualizzato.

La cartella predefinita per il salvataggio è C:\Programmi\Omron\CX-One\FBL.
Può essere modificato selezionando l'opzione "Cartella di archiviazione della libreria FB" di CX-Programmer.
La libreria FB di OMRON è disponibile nella cartella omronlib.
Creare una cartella in modo tale da poterla classificare con facilità, ad esempio Userlib\DVD.

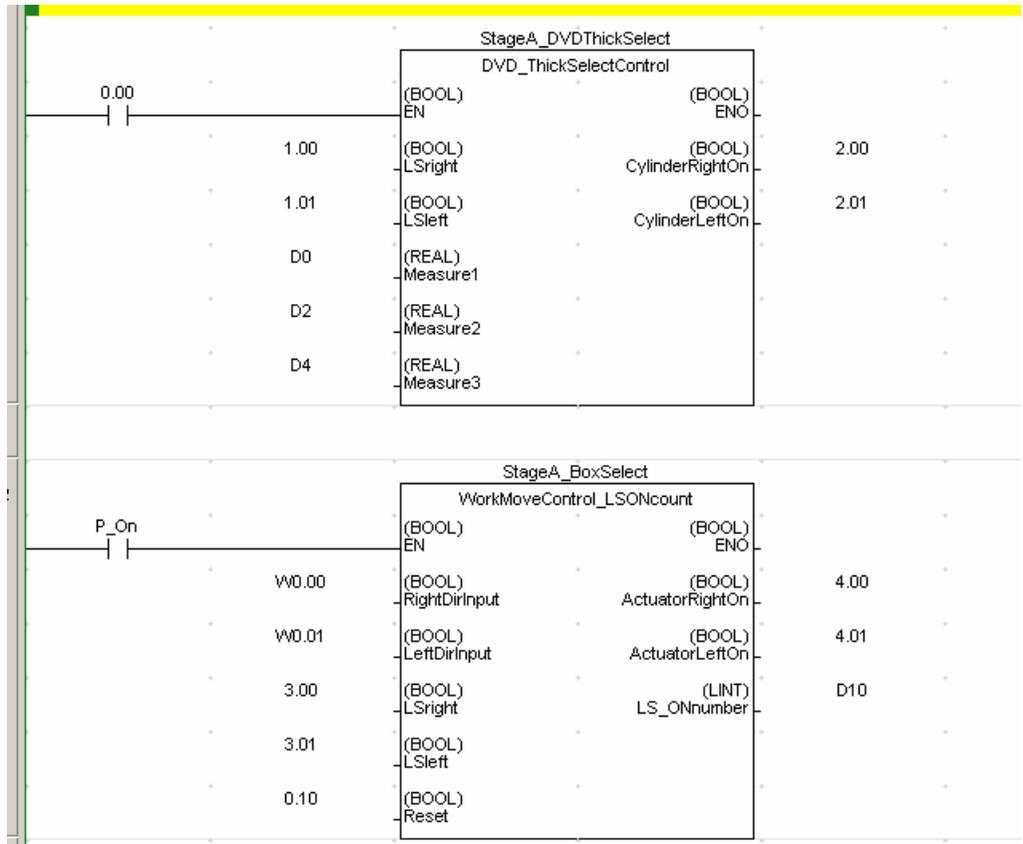
Durante il salvataggio della definizione FB che richiama un altro blocco funzione, vengono salvate entrambe le definizioni FB. Durante il recupero di un progetto, il rapporto fra elemento di chiamata e quello chiamante viene mantenuto così come è stato salvato.
La gestione della definizione FB risulta più semplice poiché la definizione FB salvata è integrata.

7. Immissione del programma principale

Aggiungere il programma principale a un file del progetto che contiene la definizione FB per la quale è stato eseguito il debug. Il programma da immettere è quello descritto a pagina 4-5. Per informazioni sulla descrizione del programma generale, vedere a pagina 5-7.

[Variabili globali]

Name	Data Type	Address / Value	Rack Location
StageA_BoxSelect	FB [WorkMoveControl_LSONcount]	N/A [Auto]	
StageA_DVDThickSelect	FB [DVD_ThickSelectControl]	N/A [Auto]	

* Le variabili di istanze diverse da quelle utilizzate per il blocco funzione vengono omesse.

Per informazioni sull'immissione di un programma, vedere le pagine da 2-6 a 2-9.

8. Debug del programma principale

Eseguire il debug del programma principale tenendo a mente i seguenti elementi:

- Le aree di programma non correlate al blocco funzione
- Le aree di programma correlate a un parametro di ingresso del blocco funzione
- Le aree di programma che fanno riferimento a un parametro di uscita del blocco funzione

Il programma principale di questo esempio non include tali aree e pertanto la relativa spiegazione viene omessa.

Eliminazione delle definizioni del blocco funzione non utilizzate

Quando si eliminano le definizioni del blocco funzione non utilizzate, non è sufficiente eliminare l'istruzione di chiamata del blocco funzione. Ciò è dovuto al fatto che le definizioni dell'istanza del blocco funzione vengono registrate nella tabella dei simboli globali.
In questo caso, una volta eseguita la compilazione (controllo del programma), le istanze del blocco funzione non utilizzate verranno visualizzate nella finestra dei risultati. In questo modo, è possibile identificare ed eliminare con facilità le definizioni delle istanze del blocco funzione non utilizzate.

Le definizioni e le istanze del blocco funzione fanno parte del programma nella CPU anche se non vengono richiamate. Si consiglia pertanto di eliminare le definizioni e le istanze del blocco funzione non utilizzate prima di trasferire il programma alla CPU.

Risultato della compilazione

Allocazione di memoria per i blocchi funzione

Per eseguire i blocchi funzione, è necessario allocare la memoria necessaria a ciascuna istanza del blocco funzione.
CX-Programmer alloca la memoria in modo automatico in base alle informazioni della seguente finestra di dialogo.
(menu PLC ® Memoria blocco funzione ® Allocazione memoria blocco funzione)
Sono disponibili 4 tipi di aree: "Non mantenere", "Mantieni", "Temporizzatori" e "Contatori". Se necessario, modificare le impostazioni.

- ☟⑥❶♋ ⑨♌③♋⓿⓪❷♋ ♋③③♋ ④⑥♒⓪♐⓪♏♋ ♒♌③③♌ ⓪④⑦⑥⑩❶♋❻⓪⑥⑤⓪
  Se si modifica l'area "Non mantenere" o quella "Mantieni", tenere in considerazione le aree di memoria allocata al modulo ITO speciale
  e all'unità CPU SIO.

- ✒⑨♌♋ ♒⓪ ④♌④⑥⑨⓪♋ ⑩⑦♌♏⓪♋③♌ ♒♌⓪ ♏③⑥♏♏♐⓪ ♐❶⑤❻⓪⑥⑤♌
  CPU CS1/CJ1-H/CJ1M (versione: 3.0 o successiva) dispongono di un'area di memoria speciale, ovvero un'area di memorizzazione (H) estesa.
  L'indirizzo dell'area va da H512 a H1535. Quest'area viene impostata in modo predefinito da CX-Programmer.
  Questa area non può essere utilizzata per gli operandi del comando ladder.

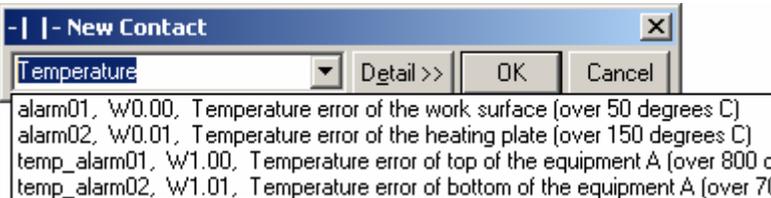
FB Instance Area	Start Address	End Address	Size
Non Retain	H512	H1407	896
Retain	H1408	H1535	128
Timers	T3072	T4095	1024
Counters	C3072	C4095	1024

Funzioni utili

Ricerca automatica degli operandi dei comandi e visualizzazione dell'elenco

Durante l'immissione degli operandi dei comandi, è possibile visualizzare automaticamente un elenco di nomi di simboli o di commenti I/O.
Durante l'immissione dell'operando del contatto, dell'uscita o di istruzioni speciali, immettere una stringa. L'elenco a discesa viene automaticamente aggiornato per visualizzare i nomi dei simboli o i commenti I/O che utilizzano la stringa specificata. Selezionando la voce dall'elenco si definiscono le informazioni sull'operando.
Si tratta di un metodo efficiente di immissione nel ladder delle informazioni sui simboli registrati.

Esempio: Immettere il testo "Temperature" nel campo di modifica della finestra di dialogo dell'operando.

Fare cl u o premere il tasto [F4]; vengono visualizzati nell'elenco tutti i simboli/gli indirizzi con il commento I/O che include il testo "temperature". Vedere di seguito:

Ad esempio, selezionare dall'elenco "temp_alarm01, W1.00, Errore temperatura della parte superiore della MacchinaA". L'operando viene impostato in modo da utilizzare il simbolo "alarm01".

Funzioni di protezione FB

È possibile adottare misure preventive impostando una password nella definizione di blocco funzione allocata al file del progetto per impedire l'utilizzo, la fuga di informazioni sul know-how del progetto, modifiche non autorizzate e alterazioni.

- ☒ꝏ⑨⓪❶❶❶⑨♋▯⊗⓪⑩❶♋③⓪❻❻♋❻⓪⓪⑤♎ ♌⓪⑩♋ꝏ⓪③⓪❶♋❶♋

Impostando il tipo di protezione "Scrittura/Visualizzazione disabilitata" si impedisce la visualizzazione del contenuto della definizioni del blocco funzione corrispondente. Applicando una protezione tramite password alla definizione del blocco funzione, è possibile impedire le fughe di informazioni relative al know-how del programma.

- ☒⑥③⑥ ⑩ꝏ⑨⓪❶❶❶⑨♋ ♌⓪⑩♋ꝏ⓪③⓪❶♋❶♋

Impostando il tipo di protezione "Solo scrittura disabilitata" si impedisce la scrittura o la modifica del contenuto della definizioni del blocco funzione corrispondente. Applicando una protezione tramite password alla definizione del blocco funzione, si impedisce che vengano apportate modifiche non autorizzate al programma.

Esempi di utilizzo dell'istruzione IF

```
IF espressione1 THEN elenco-istruzioni1
[ ELSIF espressione2 THEN elenco-istruzioni2 ]
[ ELSE elenco-istruzioni3 ]
END_IF;
```

Le espressioni espressione1 ed espressione2 devono dare come risultato un valore booleano. L'elenco-istruzioni è un elenco di istruzioni semplici, ad esempio a:=a+1; b:=3+c; e così via.

La parola chiave IF esegue l'elenco-istruzioni1 se il valore dell'espressione1 è True. Se è presente ELSIF e il valore dell'espressione1 è False e quello dell'espressione2 è True, esegue l'elenco-istruzioni2; se è presente ELSE e se il valore dell'espressione1 o dell'espressione2 è False, esegue l'elenco-istruzioni3. Dopo l'esecuzione dell'elenco-istruzioni1, dell'elenco-istruzioni2 o dell'elenco-istruzioni3, il controllo passa all'istruzione successiva a END_IF.

Possono essere presenti diverse istruzioni ELSIF all'interno di un'istruzione IF, ma solo un'istruzione ELSE.

Le istruzioni IF possono essere nidificate all'interno di altre istruzioni IF (fare riferimento all'esempio 5).

Esempio 1

```
 IF a > 0 THEN
   b := 0;
END_IF;
```

In questo esempio, se la variabile "a" è maggiore di 0, alla variabile "b" viene assegnato il valore 0.

Se la variabile "a" non è maggiore di 0, non viene eseguita alcuna azione sulla variabile "b" e il controllo viene trasferito ai passi del programma successivi alla clausola END_IF.

Esempio 2

```
 IF a THEN
   b := 0;
END_IF;
```

In questo esempio, se il valore della variabile "a" è True, alla variabile "b" viene assegnato il valore 0.

Se il valore della variabile "a" è False, non viene eseguita alcuna azione sulla variabile "b" e il controllo viene trasferito ai passi del programma successivi alla clausola END_IF.

Esempio 3

```
IF a > 0 THEN
   b := TRUE;
ELSE
   b := FALSE;
END_IF;
```

In questo esempio, se la variabile "a" è maggiore di 0, alla variabile "b" viene assegnato il valore True (1) e il controllo viene trasferito ai passi del programma successivi alla clausola END_IF.

In questo esempio, se la variabile "a" non è maggiore di 0, non viene eseguita alcuna azione sulla variabile "b" e il controllo viene trasferito all'istruzione successiva alla clausola ELSE. Alla variabile "b" verrà assegnato il valore False (0).

Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_IF.

Esempio 4

```
IF a < 10 THEN
   b := TRUE;
   c := 100;
ELSIF a  > 20 THEN
   b := TRUE;
   c := 200;
ELSE
   b := FALSE;
   c := 300;
END_IF;
```

In questo esempio, se il valore della variabile "a" è minore di 10, alla variabile "b" viene assegnato il valore True (1) e alla variabile "c" viene assegnato il valore 100. Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_IF.

Se la variabile "a" è uguale o maggiore di 10, il controllo passa alla clausola ELSE_IF. Se la variabile "a" è maggiore di 20, alla variabile "b" viene assegnato il valore True (1) e alla variabile "c" il valore 200. Il controllo viene trasferito ai passi del programma successivi alla clausola END_IF.

Se la variabile "a" è compresa tra i valori 10 e 20 (ossia entrambe le condizioni precedenti di IF ed ELSE_IF sono false), il controllo passa alla clausola ELSE e alla variabile "b" viene assegnato il valore False (0) e alla variabile "c" il valore 300. Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_IF.

Esempi di utilizzo dell'istruzione IF

Esempio 5

```
IF a THEN
  b := TRUE;
ELSE
  IF c>0 THEN
    d := 0;
  ELSE
    d := 100;
  END_IF;
  d := 400;
END_IF;
```

In questo esempio di istruzione IF .. THEN nidificata, se il valore della variabile "a" è True (1), alla variabile "b" viene assegnato il valore True (1) e il controllo viene trasferito ai passi del programma successivi alla clausola END_IF associata.

Se il valore di "a" è False (0), non viene eseguita alcuna azione sulla variabile "b" e il controllo viene trasferito all'istruzione successiva alla clausola ELSE (in questo esempio, un'altra istruzione IF .. THEN, che viene eseguita come descritto nell'esempio 3, benché sia possibile utilizzare una qualsiasi delle istruzioni IEC61131-3 supportate).

Dopo l'esecuzione dell'istruzione IF .. THEN, alla variabile "d" viene assegnato il valore 400.

Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_IF.

Esempi di utilizzo dell'istruzione WHILE

```
WHILE espressione DO
  elenco-istruzioni;
END_WHILE;
```

L'espressione WHILE deve dare come risultato un valore booleano. L'elenco-istruzioni è un elenco di istruzioni semplici.

La parola chiave WHILE esegue ripetutamente l'elenco-istruzioni se il valore dell'espressione è True. Se il valore dell'espressione è False, il controllo viene trasferito all'istruzione successiva a END_WHILE.

Esempio 1

```
WHILE a < 10 DO
  a := a + 1;
  b := b * 2.0;
END_WHILE;
```

In questo esempio, viene valutata l'espressione WHILE e se il relativo valore è True (ossia se la variabile "a" è minore di 10) viene eseguito l'elenco-istruzioni (a:= a+1; e b:=b*2.0;). Dopo l'esecuzione dell'elenco-istruzioni, il controllo torna all'inizio dell'espressione WHILE. Questo processo viene ripetuto finché la variabile "a" è minore di 10. Se la variabile "a" è maggiore o uguale a 10, l'elenco-istruzioni non viene eseguito e il controllo viene trasferito ai passi del programma successivi alla clausola END_WHILE.

Esempio 2

```
WHILE a DO
  b := b + 1;
  IF b > 10 THEN
    a:= FALSE;
  END_IF;
END_WHILE;
```

In questo esempio, viene valutata l'espressione WHILE e se il relativo valore è True (ossia se il valore della variabile "a" è True) viene eseguito l'elenco-istruzioni (b := b + 1; e l'istruzione IF .. THEN). Dopo l'esecuzione dell'elenco-istruzioni, il controllo torna all'inizio dell'espressione WHILE. Questo processo viene ripetuto finché il valore della variabile "a" rimane True. Quando il valore della variabile "a" diventa False, l'elenco-istruzioni non viene più eseguito e il controllo viene trasferito ai passi del programma successivi alla clausola END_WHILE.

Esempio 3

```
WHILE (a + 1) >= (b * 2) DO
  a := a + 1;
  b := b / c;
END_WHILE;
```

In questo esempio, viene valutata l'espressione WHILE e se il relativo valore è True (ossia se la variabile "a" più 1 è maggiore o uguale alla variabile "b" moltiplicata per 2), viene eseguito l'elenco-istruzioni (a:=a+1; e b:= b/c;). Dopo l'esecuzione dell'elenco-istruzioni, il controllo torna all'inizio dell'espressione WHILE. Questo processo viene ripetuto finché il valore dell'espressione WHILE è True. Quando il valore dell'espressione WHILE diventa False, l'elenco-istruzioni non viene più eseguito e il controllo viene trasferito ai passi del programma successivi alla clausola END_WHILE.

Esempi di utilizzo dell'istruzione WHILE

Esempio 4

```
WHILE (a - b) <= (b + c) DO
   a := a + 1;
   b := b * a;
END_WHILE;
```

In questo esempio, viene valutata l'espressione WHILE e se il relativo valore è True (ossia la variabile "a" meno la variabile "b" è minore o uguale alla variabile "b" più la variabile "c"), viene eseguito l'elenco-istruzioni (a:=a+1; e b := b*a;). Dopo l'esecuzione dell'elenco-istruzioni, il controllo torna all'inizio dell'espressione WHILE. Questo processo viene ripetuto finché il valore dell'espressione WHILE è True. Quando il valore dell'espressione WHILE diventa False, l'elenco-istruzioni non viene più eseguito e il controllo viene trasferito ai passi del programma successivi alla clausola END_WHILE.

Esempi di utilizzo dell'istruzione REPEAT

```
REPEAT
   elenco-istruzioni;
UNTIL espressione
END_REPEAT;
```

L'espressione REPEAT deve dare come risultato un valore booleano. L'elenco-istruzioni è un elenco di istruzioni semplici.

La parola chiave REPEAT esegue ripetutamente l'elenco-istruzioni se il valore dell'espressione è False. Quando il valore dell'espressione diventa True, il controllo passa all'istruzione successiva a END_REPEAT.

Esempio 1

```
REPEAT
   a := a + 1;
   b := b * 2.0;
UNTIL a > 10
END_REPEAT;
```

In questo esempio, viene eseguito l'elenco-istruzioni (a:=a+1 e b:= b* 2.0;). Dopo l'esecuzione dell'elenco-istruzioni viene valutata l'espressione UNTIL e se il relativo valore è False (ossia se la variabile "a" è minore o uguale a 10), il controllo torna all'inizio dell'espressione REPEAT e viene nuovamente eseguito l'elenco-istruzioni. Questo processo viene ripetuto finché il valore dell'espressione UNTIL è False. Quando il valore dell'espressione UNTIL diventa True (ossia se la variabile "a" è maggiore di 10), il controllo viene trasferito ai passi del programma successivi alla clausola END_REPEAT.

Esempio 2

```
REPEAT
   b := b + 1;
   IF b > 10 THEN
      a:= FALSE;
   END_IF;
UNTIL a
END_REPEAT;
```

In questo esempio, viene eseguito l'elenco-istruzioni (b:= b+1 e l'istruzione IF .. THEN). Dopo l'esecuzione dell'elenco-istruzioni viene valutata l'espressione UNTIL e se il relativo valore è False (ossia se il valore della variabile "a" è False), il controllo torna all'inizio dell'espressione REPEAT e viene nuovamente eseguito l'elenco-istruzioni. Questo processo viene ripetuto finché il valore dell'espressione UNTIL è False. Quando il valore dell'espressione UNTIL diventa True (ossia se il valore della variabile "a" è True), il controllo viene trasferito ai passi del programma successivi alla clausola END_REPEAT.

Esempio 3

```
REPEAT
   a := a + 1;
   b := b / c;
UNTIL (a + 1) >= (b * 2)
END_REPEAT;
```

In questo esempio, viene eseguito l'elenco-istruzioni (a:= a+1; e b:= b/c;). Dopo l'esecuzione dell'elenco-istruzioni viene valutata l'espressione UNTIL e se il relativo valore è False (ossia se la variabile "a" più 1 è minore della variabile "b" moltiplicata per 2), il controllo torna all'inizio dell'espressione REPEAT e viene nuovamente eseguito l'elenco-istruzioni. Questo processo viene ripetuto finché il valore dell'espressione UNTIL è False. Quando il valore dell'espressione UNTIL diventa True (ossia se la variabile "a" più 1 è maggiore o uguale alla variabile "b" moltiplicata per 2), il controllo viene trasferito ai passi del programma successivi alla clausola END_REPEAT.

Esempio 4

```
REPEAT
   a := a + 1;
   b := b * a;
UNTIL (a - b) <= (b + c)
END_REPEAT;
```

In questo esempio, viene eseguito l'elenco-istruzioni (a:= a+1; e b:= b*a;). Dopo l'esecuzione dell'elenco-istruzioni viene valutata l'espressione UNTIL e se il relativo valore è False (ossia se la variabile "a" meno la variabile "b" è maggiore della variabile "b" più la variabile "c"), il controllo torna all'inizio dell'espressione REPEAT e viene nuovamente eseguito l'elenco-istruzioni. Questo processo viene ripetuto finché il valore dell'espressione UNTIL è False. Quando il valore dell'espressione UNTIL diventa True (ossia se la variabile "a" meno la variabile "b" è minore o uguale alla variabile "b" più la variabile "c"), il controllo viene trasferito ai passi del programma successivi alla clausola END_REPEAT.

Esempi di utilizzo dell'istruzione FOR

FOR variabile di controllo := espressione intera1 TO espressione intera2 [BY espressione intera3] DO

elenco-istruzioni;

END_FOR;

La variabile di controllo FOR deve essere di tipo intero. Le espressioni intere FOR devono dare come risultato un valore dello stesso tipo intero della variabile di controllo. L'elenco-istruzioni è un elenco di istruzioni semplici.

La parola chiave FOR esegue ripetutamente l'elenco-istruzioni se la variabile di controllo è compresa nell'intervallo tra espressione intera1 ed espressione intera2. Se è presente la clausola BY, la variabile di controllo viene incrementata del valore prodotto dall'espressione intera3, altrimenti viene incrementata di 1 per impostazione predefinita. La variabile di controllo viene incrementata dopo l'esecuzione di ogni chiamata dell'elenco-istruzioni. Quando la variabile di controllo non è più compresa nell'intervallo tra l'espressione intera1 e l'espressione intera2, il controllo passa all'istruzione successiva a END_FOR.

Le istruzioni FOR possono essere nidificate all'interno di altre istruzioni FOR.

Esempio 1

```
FOR a := 1 TO 10 DO
  b := b + a;
END_FOR;
```

In questo esempio, l'espressione FOR viene inizialmente valutata e la variabile "a" viene inizializzata con il valore 1. Il valore della variabile "a" viene quindi confrontato con il valore "TO" dell'istruzione FOR e se è minore o uguale a 10 viene eseguito l'elenco-istruzioni (in questo esempio, b:= b+a;). La variabile "a" verrà quindi incrementata di 1 e il controllo tornerà all'inizio dell'istruzione FOR. La variabile "a" verrà nuovamente confrontata con il valore "TO" e se è minore o uguale a 10 verrà eseguito nuovamente l'elenco-istruzioni. Questo processo viene ripetuto finché il valore della variabile "a" non risulta maggiore di 10, quindi il controllo viene trasferito ai passi del programma successivi alla clausola END_FOR.

Esempio 2

```
FOR a := 1 TO 10 BY 2 DO
  b := b + a;
  c := c + 1.0;
END_FOR;
```

In questo esempio, l'espressione FOR viene inizialmente valutata e la variabile "a" viene inizializzata con il valore 1. Il valore della variabile "a" viene quindi confrontato con il valore "TO" dell'istruzione FOR e se è minore o uguale a 10 viene eseguito l'elenco-istruzioni (in questo esempio, b:=b+a; e c:=c+1.0;). La variabile "a" verrà quindi incrementata di 2 e il controllo tornerà all'inizio dell'istruzione FOR. La variabile "a" verrà nuovamente confrontata con il valore "TO" e se è minore o uguale a 10 verrà eseguito nuovamente l'elenco-istruzioni. Questo processo viene ripetuto finché il valore della variabile "a" non risulta maggiore di 10, quindi il controllo viene trasferito ai passi del programma successivi alla clausola END_FOR.

Esempio 3

```
FOR a := 10 TO 1 BY -1 DO
  b := b + a;
  c := c + 1.0;
END_FOR;
```

In questo esempio, l'espressione FOR viene inizialmente valutata e la variabile "a" viene inizializzata con il valore 10. Il valore della variabile "a" viene quindi confrontato con il valore "TO" dell'istruzione FOR e se è maggiore o uguale a 1 viene eseguito l'elenco-istruzioni (in questo esempio, b:= b+a; e c:=c+1.0;). La variabile "a" verrà quindi decrementata di 1 e il controllo tornerà all'inizio dell'istruzione FOR. La variabile "a" verrà nuovamente confrontata con il valore "TO" e se è maggiore o uguale a 1 verrà eseguito nuovamente l'elenco-istruzioni. Questo processo viene ripetuto finché il valore della variabile "a" non risulta minore di 1, quindi il controllo viene trasferito ai passi del programma successivi alla clausola END_FOR.

Esempio 4

```
FOR a := b + 1 TO c + 2 DO
  d := d + a;
  e := e + 1;
END_FOR;
```

In questo esempio, l'espressione FOR viene inizialmente valutata e la variabile "a" viene inizializzata con il valore della variabile "b" + 1. Il valore "TO" dell'istruzione FOR viene valutato in base al valore della variabile "c" + 2. Il valore della variabile "a" viene quindi confrontato con il valore "TO" e se è minore o uguale a tale valore viene eseguito l'elenco-istruzioni (in questo esempio, d:=d+a; ed e:=e+1;). La variabile "a" verrà quindi incrementata di 1 e il controllo tornerà all'inizio dell'istruzione FOR. La variabile "a" verrà nuovamente confrontata con il valore "TO" e se è minore o uguale a 10 verrà eseguito nuovamente l'elenco-istruzioni. Questo processo viene ripetuto finché il valore della variabile "a" non risulta maggiore del valore "TO", quindi il controllo viene trasferito ai passi del programma successivi alla clausola END_FOR.

Esempi di utilizzo dell'istruzione FOR

Esempio 5

```
FOR a := b + c TO d - e BY f DO
  g := g + a;
  h := h + 1.0;
END_FOR;
```

In questo esempio, l'espressione FOR viene inizialmente valutata e la variabile "a" viene inizializzata con il valore della variabile "b" più la variabile "c". Il valore "TO" dell'istruzione FOR viene valutato in base al valore della variabile "d" meno la variabile "e". Il valore della variabile "a" viene quindi confrontato con il valore "TO". Se il valore della variabile "f" è positivo e il valore della variabile "a" è minore o uguale al valore "TO", viene eseguito l'elenco-istruzioni (in questo esempio, g:=g+a; e h:=h+1.0;). Se il valore della variabile "f" è negativo e il valore della variabile "a" è maggiore o uguale al valore "TO", viene eseguito l'elenco-istruzioni (in questo esempio, g:=g+a; e h:=h+1.0;). La variabile "a" verrà quindi incrementata o decrementata del valore della variabile "f" e il controllo tornerà all'inizio dell'istruzione FOR. La variabile "a" verrà nuovamente confrontata con il valore "'TO" e, se appropriato, verrà eseguito l'elenco-istruzioni (come descritto sopra).

Questo processo viene ripetuto finché il valore della variabile "a" non risulta maggiore del valore "TO" (se il valore della variabile "f" è positivo) oppure finché il valore della variabile "a" non risulta minore del valore "TO" (se il valore della variabile "f" è negativo), quindi il controllo viene trasferito ai passi del programma successivi alla clausola END_FOR.

Esempi di utilizzo dell'istruzione CASE

```
CASE espressione OF
  case etichetta1 [ , case etichetta2 ]  [ .. case etichetta3 ]  : elenco-istruzioni1;
[ ELSE
   elenco-istruzioni2 ]
END_CASE;
```

Il valore della valutazione dell'espressione CASE deve dare come risultato un numero intero. L'elenco-istruzioni è un elenco di istruzioni semplici. Le etichette devono essere valori interi letterali validi, ad esempio 0, 1, +100, -2 e così via.

La parola chiave CASE valuta l'espressione ed esegue l'elenco-istruzioni associato all'etichetta il cui valore corrisponde all'espressione iniziale. Il controllo viene quindi trasferito all'istruzione successiva a END_CASE. Se il valore dell'espressione non corrisponde a nessuna etichetta ed è presente un comando ELSE, viene eseguito l'elenco-istruzioni associato alla parola chiave ELSE. Se la parola chiave ELSE non è presente, il controllo viene trasferito all'istruzione successiva a END_CASE.

Possono essere presenti diverse istruzioni etichette (ed elenco-istruzioni associati) all'interno di un'istruzione CASE, ma solo un'istruzione ELSE.

L'operatore "," viene utilizzato per elencare più etichette associate allo stesso elenco-istruzioni..

L'operatore ".." indica un intervallo di valori per un'etichetta. Se l'espressione CASE rientra in tale intervallo, viene eseguito l'elenco-istruzioni associato. Ad esempio, l'etichetta 1..10 a:=a+1; esegue a:a+1 se l'espressione CASE è maggiore o uguale a 1 e minore di 10.

Esempio 1

```
CASE a OF
  2 :   b := 1;
  5 :   c := 1.0;
END_CASE;
```

In questo esempio l'istruzione CASE viene valutata e confrontata con ciascun valore di confronto dell'istruzione CASE (2 e 5 in questo esempio).

Se il valore della variabile "a" è 2, viene eseguito il relativo elenco-istruzioni (in questo esempio, b:=1;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" è 5, viene eseguito il relativo elenco-istruzioni (in questo esempio, c:=1.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" non corrisponde ad alcun valore di confronto dell'istruzione CASE, il controllo viene trasferito ai passi del programma successivi alla clausola END_CASE.

Esempio 2

```
CASE a + 2 OF
  -2 :  b := 1;
  5 :   c := 1.0;
ELSE
  d := 1.0;
END_CASE;
```

In questo esempio, l'istruzione CASE viene valutata e confrontata con ciascun valore di confronto dell'istruzione CASE (-2 e 5 in questo esempio).

Se il valore della variabile "a" + 2 è -2, viene eseguito il relativo elenco-istruzioni (in questo esempio, b:=1;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE. Se il valore della variabile "a" + 2 è 5, viene eseguito il relativo elenco-istruzioni (in questo esempio, c:=1.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE. Se il valore della variabile "a" + 2 è diverso da -2 o 5, viene eseguito l'elenco-istruzioni nella condizione ELSE (in questo esempio, d:= 1.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Esempi di utilizzo dell'istruzione CASE

Esempio 3

```
 CASE a + 3 * b OF
   1, 3 :    b := 2;
   7, 11 :  c := 3.0;
ELSE
   d := 4.0;
END_CASE;
```

In questo esempio l'istruzione CASE viene valutata e confrontata con ciascun valore di confronto dell'istruzione CASE (1 o 3 e 7 o 11 in questo esempio).

Se il valore della variabile "a" + 3 moltiplicato per la variabile "b" è 1 o 3, viene eseguito il relativo elenco-istruzioni (in questo esempio, b:= 2;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" + 3 moltiplicato per la variabile "b" è 7 o 11, viene eseguito il relativo elenco-istruzioni (in questo esempio, c:= 3.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" + 3 moltiplicato per la variabile "b" è diverso da 1, 3, 7 o 11, viene eseguito l'elenco-istruzioni nella condizione ELSE (in questo esempio, d:=4.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Esempio 4

```
 CASE a OF
-2, 2, 4 :     b := 2;
               c := 1.0;
6..11, 13 :    c := 2.0;
1, 3, 5 :      c := 3.0;
ELSE
   b := 1;
   c := 4.0;
END_CASE;
```

In questo esempio l'istruzione CASE viene valutata e confrontata con ciascun valore di confronto dell'istruzione CASE, (-2, 2 o 4), (da 6 a 11 o 13) e (1, 3 o 5) in questo esempio.

Se il valore della variabile "a" è –2, 2 o 4, viene eseguito il relativo elenco-istruzioni (in questo esempio, b:=2; e c:=1.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" è 6, 7, 8, 9, 10, 11 o 13, viene eseguito il relativo elenco-istruzioni (in questo esempio, c:= 2.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" è 1, 3 o 5, viene eseguito il relativo elenco-istruzioni (in questo esempio, c:=3.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Se il valore della variabile "a" è diverso da tutti i valori sopra riportati, viene eseguito l'elenco-istruzioni nella condizione ELSE (in questo esempio, b:=1; e c:=4.0;). Il controllo viene quindi trasferito ai passi del programma successivi alla clausola END_CASE.

Esempi di utilizzo dell'istruzione EXIT

```
WHILE espressione DO
   elenco-istruzioni1;
   EXIT;
END_WHILE;
elenco-istruzioni2;
```

```
REPEAT
  elenco-istruzioni1;
  EXIT;
UNTIL espressione
END_REPEAT;
elenco-istruzioni2;
```

```
FOR variabile di controllo := espressione intera1 TO espressione intera2 [ BY espressione intera3 ] DO
  elenco-istruzioni1;
  EXIT;
END_FOR;
elenco-istruzioni2;
```

L'elenco-istruzioni è un elenco di istruzioni semplici.

La parola chiave EXIT interrompe l'esecuzione del ciclo di ripetizione per passare all'istruzione successiva e può essere utilizzata solo nelle istruzioni ripetitive (istruzioni WHILE, REPEAT, FOR). Quando la parola chiave EXIT viene eseguita dopo l'elenco-istruzioni1 nel ciclo ripetitivo, il controllo viene trasferito immediatamente all'elenco-istruzioni2.

Esempio 1

```
WHILE a DO
 IF c = TRUE THEN
   b:=0;EXIT;
 END_IF;
 IF b > 10 THEN
    a:= FALSE;
  END_IF;
END_WHILE;
d:=1;
```

Se il valore della prima espressione IF è True (ossia se il valore della variabile "c" è True), l'elenco-istruzioni (b:=0; ed EXIT;) viene eseguito durante l'esecuzione del ciclo WHILE. Una volta eseguita la parola chiave EXIT, il ciclo WHILE viene interrotto e il controllo viene trasferito IF (d:=1) successiva alla clausola END_WHILE.

Esempio 2

```
 a:= FALSE;
FOR i:=1 TO 20 DO
     FOR j:=0 TO 9 DO
            IF i>=10 THEN
                    n:=i*10+j;
                    a:=TRUE;EXIT;
            END_IF;
     END_FOR;
     IF a THEN EXIT; END_IF;
END_FOR;
d:=1;
```

Se il valore della prima espressione IF è True (ovvero se il valore di i>=10 è True) nel ciclo interno FOR, l'elenco-istruzioni (n:=i*10+j; a:=TRUE; ed EXIT;) viene eseguito durante l'esecuzione del ciclo FOR. Una volta eseguita la parola chiave EXIT, il ciclo interno FOR viene interrotto e il controllo viene trasferito all'istruzione IF successiva alla clausola END_FOR. Se il valore di questa espressione IF è True (ossia se il valore della variabile "a" è True), viene eseguita la parola chiave EXIT, il ciclo esterno FOR viene interrotto dopo la clausola END_FOR e il controllo viene trasferito all'istruzione successiva (d:=1;).

Esempi di utilizzo dell'istruzione RETURN

elenco-istruzioni1;
RETURN;
elenco-istruzioni2;

L'elenco-istruzioni è un elenco di istruzioni semplici.

La parola chiave RETURN interrompe l'esecuzione all'interno del blocco funzione dopo l'elenco-istruzioni1, quindi il controllo torna al programma che ha eseguito la chiamata al blocco funzione senza eseguire l'elenco-istruzioni2.

Esempio 1

```
IF a_1*b>100 THEN
  c:=TRUE;RETURN;
END_IF;
IF a_2*(b+10)>100 THEN
  c:=TRUE;RETURN;
END_IF;
IF a_3*(b+20)>100 THEN
  c:=TRUE;
END_IF;
```

Se il valore della prima o della seconda istruzione IF è True (ossia se "a_1*b" o "a_2*(b + 10)" è maggiore di 100), viene eseguita l'istruzione (c := TRUE; e RETURN;). L'esecuzione della parola chiave RETURN interrompe l'esecuzione all'interno del blocco funzione e il controllo torna al programma che ha eseguito la chiamata al blocco funzione.

Esempi di matrici

nome variabile [indice]

Una matrice è una raccolta di variabili dello stesso tipo. Le dimensioni di una matrice possono essere definite nella tabella variabili del blocco funzione.

È possibile accedere a una variabile utilizzando l'operatore pedice [ ] della matrice.

L'indice consente di accedere a una specifica variabile all'interno di una matrice. L'indice deve essere un valore letterale positivo, un'espressione intera o una variabile intera. L'indice parte da zero, ossia specificando 0 come indice si accede alla prima variabile, specificando 1 come indice si accede alla seconda variabile e così via.

Avviso

Se l'indice è costituito da un'espressione o una variabile intera, è necessario assicurarsi che il valore dell'indice risultante rientri nell'intervallo di indice valido della matrice. È importante evitare di accedere a una matrice specificando un indice errato. Per ulteriori informazioni su come scrivere codice corretto quando si utilizzano offset per matrici di variabili, fare riferimento all'esempio 5.

Esempio 1

```
a[0] := 1;
a[1] := -2;
a[2] : = 1+2;
a[3] : = b;
a[4] : = b+1;
```

In questo esempio la variabile "a" è una matrice di 5 elementi di tipo INT. Anche la variabile "b" è di tipo INT. In fase di esecuzione, il primo elemento della matrice viene impostato su 1, il secondo su –2, il terzo su 3 (ossia 1+2), il quarto sul valore della variabile "b" e il quinto sul valore della variabile "b" più 1.

Esempio 2

```
c[0] := FALSE;
c[1] := 2>3;
```

In questo esempio la variabile "c" è una matrice di 2 elementi di tipo BOOL. In fase di esecuzione, il primo e il secondo elemento della matrice vengono impostati su False (perché l'espressione 2 maggiore di 3 risulta falsa).

Esempi di matrici

Esempio 3

```
d[9]:= 2,0;
```

In questo esempio la variabile "d" è una matrice di 10 elementi di tipo REAL. In fase di esecuzione, l'ultimo elemento della matrice (il decimo elemento) viene impostato su 2.0.

Esempio 4

```
a[1] := b[2];
```

In questo esempio le variabili "a" e "b" sono matrici con lo stesso tipo di dati. In fase di esecuzione, il secondo elemento nella variabile "a" viene impostato sul valore del terzo elemento nella variabile "b".

Esempio 5

```
a[b] := 1;
a[b+1] := 1;
a[(b+c) *( d-e)] := 1;
```

Nota: poiché per accedere alla matrice vengono utilizzate espressioni e variabili intere, il valore effettivo dell'indice non sarà noto fino al momento dell'esecuzione. È quindi necessario assicurarsi che l'indice sia compreso nell'intervallo valido della matrice a. Ad esempio, per maggiore sicurezza è possibile verificare la validità dell'indice della matrice come descritto di seguito:

```
f := (b+c) *( d-e);
IF (f >0) AND (f<5) THEN
   a[f] := 1;
END_IF;
```

Dove la variabile "f" è di tipo INT.

Esempio 6

```
a[b[1]]:= c;
a[b[2] + 3]:= c;
```

Questo esempio mostra come utilizzare un'espressione di elementi di matrice all'interno di un'altra espressione di elementi di matrice.

Funzioni numeriche e aritmetiche

Funzione	Descrizione	Tipo di dati argomento	Tipo valore restituito	Operazione	Esempio
ABS(argomento)	Valore assoluto	INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL	NT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL	| argomento |	a:=ABS(b)
SQRT(argomento)	Radice quadrata	REAL, LREAL	REAL, LREAL		a:=SQRT(b)
LN(argomento)	Logaritmo naturale	REAL, LREAL	REAL, LREAL		a:=LN(b)
LOG(argomento)	Logaritmo comune	REAL, LREAL	REAL, LREAL		a:=LOG(b)
EXP(argomento)	Esponenziale naturale	REAL, LREAL	REAL, LREAL		a:=EXP(b)
SIN(argomento)	Seno	REAL, LREAL	REAL, LREAL	SIN(argomento)	a:=SIN(b)
COS(argomento)	Coseno	REAL, LREAL	REAL, LREAL	COS(argomento)	a:=COS(b)
TAN(argomento)	Tangente	REAL, LREAL	REAL, LREAL	TAN(argomento)	a:=TAN(b)
ASIN(argomento)	Arcoseno	REAL, LREAL	REAL, LREAL		a:=ASIN(b)
ACOS(argomento)	Arcocoseno	REAL, LREAL	REAL, LREAL		a:=ACOS(b)
ATAN(argomento)	Arcotangente	REAL, LREAL	REAL, LREAL		a:=ATAN(b)
EXPT(base, esponente)	Esponenziale	Base: REAL, LREAL Esponente: INT, DINT, LINT, UINT, UDINT, ULINT	REAL, LREAL		a:=EXPT(b, c)

Garanzia e considerazioni sull'applicazione

Leggere attentamente e comprendere

Prima di procedere all'acquisto dei prodotti il cliente si assume l'onere di leggere attentamente e comprendere questo documento. Per eventuali domande o commenti, rivolgersi all'ufficio OMRON di competenza.

Garanzia e limitazione di responsabilità

GARANZIA

OMRON garantisce i propri prodotti da difetti di materiali e/o vizi di costruzione per un periodo di un anno (o per altro periodo se specificato) dalla data di consegna. L'onere della prova del difetto è a carico dell'acquirente. La garanzia si limita alla riparazione del prodotto o, a giudizio insindacabile di OMRON, alla sua sostituzione.

OMRON NON RICONOSCE ALTRA GARANZIA, ESPLICITA O IMPLICITA, COMPRESE IN VIA ESEMPLIFICATIVA QUELLE DI NON-VIOLAZIONE, DI COMMERCIABILITA' E DI IDONEITA' A FINI PARTICOLARI. L'ACQUIRENTE O L'UTILIZZATORE RICONOSCE LA PROPRIA ESCLUSIVA RESPONSABILITA' NELL'AVER DETERMINATO L'IDONEITA' DEL PRODOTTO A SODDISFARE I REQUISITI IMPLICITI NELL'USO PREVISTO DELLO STESSO.

LIMITAZIONE DI RESPONSABILITA'

OMRON NON SARÀ RESPONSABILE DEI DANNI, DELLE PERDITE DI PROFITTO O DELLE PERDITE COMMERCIALI SPECIALI, INDIRETTE O EMERGENTI IN QUALUNQUE MODO RICONDUCIBILI AI PRODOTTI, ANCHE QUANDO LE RICHIESTE DI INDENNIZZO POGGINO SU CONTRATTO, GARANZIA, NEGLIGENZA O RESPONSABILITÀ INCONDIZIONATA.

In nessun caso la responsabilità di OMRON potrà superare il prezzo del singolo prodotto in merito al quale sia stata definita la responsabilità.

IN NESSUN CASO OMRON SARÀ RESPONSABILE PER GARANZIA, RIPARAZIONE O ALTRA RICHIESTA DI INDENNIZZO RELATIVA AI PRODOTTI SE L'ANALISI, CONDOTTA DA OMRON, NON CONFERMERÀ CHE I PRODOTTI SONO STATI CORRETTAMENTE UTILIZZATI, IMMAGAZZINATI, INSTALLATI E SOTTOPOSTI A MANUTENZIONE, E CHE NON SONO STATI OGGETTO DI CONTAMINAZIONI, ABUSI, USI IMPROPRI, MODIFICHE O RIPARAZIONI DA PARTE DI CENTRI NON AUTORIZZATI DA OMRON.

Considerazioni sull'applicazione

IDONEITA' ALL'USO PREVISTO

OMRON non sarà responsabile della conformità a normative, regolamenti e leggi applicabili a combinazioni di prodotti nell'applicazione del cliente o nell'impiego dei prodotti stessi. Il cliente e/o l'utilizzatore hanno la responsabilità di adottare tutte le misure necessarie a determinare l'idoneità del prodotto ai sistemi, ai macchinari e alle apparecchiature con i quali verrà utilizzato. Il cliente e/o l'utilizzatore hanno la responsabilita' di conoscere ed osservare tutte le proibizioni, regole, limitazioni e divieti applicabili all'uso del prodotto e/o al prodotto stesso.

NON UTILIZZARE MAI I PRODOTTI IN APPLICAZIONI CHE IMPLICHINO GRAVI RISCHI PER L'INCOLUMITÀ DELLE PERSONE O DI DANNI ALLA PROPRIETÀ SENZA PRIMA AVERE APPURATO CHE L'INTERO SISTEMA SIA STATO PROGETTATO TENENDO IN CONSIDERAZIONE TALI RISCHI E CHE I PRODOTTI OMRON SIANO STATI VALUTATI, INSTALLATI E PROVATI CORRETTAMENTE IN VISTA DELL'USO AL QUALE SONO DESTINATI NELL'AMBITO DELL'APPARECCHIATURA O DEL SISTEMA.

Dichiarazione di non responsabilità

DATI SULLE PRESTAZIONI

I dati sulle prestazioni forniti in questo catalogo non costituiscono una garanzia, bensì solo una guida alla scelta delle soluzioni più adeguate alle esigenze dell'utente. Essendo il risultato delle condizioni di collaudo di OMRON, tali dati devono essere messi in relazione agli effettivi requisiti di applicazione. Le prestazioni effettive sono soggette alla *Garanzia e Limitazione di Responsabilità* di OMRON.

MODIFICHE ALLE SPECIFICHE

Le caratteristiche e gli accessori del prodotto possono essere soggetti a modifiche a scopo di perfezionamento o per altri motivi. Per confermare le caratteristiche effettive del prodotto acquistato, rivolgersi all'ufficio OMRON di competenza.

DIMENSIONI E PESI

Pesi e misure sono nominali e non devono essere utilizzati in progettazione o produzione, anche quando sono indicati i valori di tolleranza.

Cat. No. R133-IT2-02

Le informazioni contenute nel presente documento sono soggette a modifiche senza preavviso.

ITALIA
Omron Electronics SpA
Viale Certosa, 49 - 20149 Milano
Tel: +39 02 32 681
Fax: +39 02 32 68 282
www.omron.it

Nord Ovest Tel: +39 02 326 88 00
Milano Tel: +39 02 326 87 77
Bologna Tel: +39 051 613 66 11
Terni Tel: +39 074 45 45 11

SVIZZERA
Omron Electronics AG
Sennweidstrasse 44, CH-6312 Steinhausen
Tel: +41 (0) 41 748 13 13
Fax: +41 (0) 41 748 13 45
www.omron.ch

Romanel Tel: +41 (0) 21 643 75 75